

# დაქირავებულ პირთა რეესტრი

## 1.0.1



# დაქირავებულ პირთა რეესტრი

1.0.1

[ ზოგადი ლინკი: <https://eapi.rs.ge/> ]

## რეზიუმე

დოკუმენტის მიზანია აღწეროს ელექტრონული ურთიერთობის პროცესი, ელ.

ურთიერთობის ტექნიკური მოთხოვნები, მიღებული და შეთანხმებული იქნას ელექტრონული ურთიერთობების ტექნიკური სტანდარტი, პროტოკოლი და პირობები.

სერვისთან დაკავშირებული ნებისმიერი კითხვის ან პრობლემის შემთხვევაში დაგვიკავშირდით: [api-support@rs.ge](mailto:api-support@rs.ge)

მომზადებულია სსიპ შემოსავლების სამსახურის  
საინფორმაციო ტექნოლოგიების ცენტრის მიერ  
იანვარი 2022 წელი

## შინაარსი

ცვლილებების ისტორია.....	4
განმარტებები .....	5
Eservice API .....	6
შეცდომები .....	7
ავტორიზაცია .....	8
სისტემიდან გამოსვლა .....	13
დაქირავებულ პირთა რეესტრი .....	14
დაქირავებული პირის მონაცემები ID-ის მიხედვით.....	15
დაქირავებული პირის შენახვა .....	16
დაქირავებულ პირთა სია .....	18
ქვეყნების ჩამონათვალი.....	22

## ცვლილებების ისტორია

ვერსია	თარიღი	შენიშვნა
1.0.0	17-01-2022	დოკუმენტის პირველი ვერსია
1.0.1	30-07-2024	დაემატა MAXIMUM_ROWS პარამეტრი ListEmployees მეთოდში

## განმარტებები

შემოსავლების სამსახური - სსიპ შემოსავლების სამსახური;

მომხმარებელი - შემოსავლების სამსახურის მონაცემთა ბაზაში გადასახადის გადამხდელად რეგისტრირებული პირი, რომელიც ამავე დროს არის ელექტრონული დეკლარირების აქტიური მომხმარებელი. სავალდებულო არაა პირი იყოს დღგ-ს გადამხდელად რეგისტრირებული;

აქტიური - პირს აქტიური სტატუსი ენიჭება იმ შემთხვევაში თუ მასთან არსებობს შრომითი ხელშეკრულება;

შეჩერებული - პირს სტატუსი შეჩერებული ენიჭება შრომის კოდექსის 36-ე მუხლის მე-2 ნაწილით გათვალისწინებული საფუძვლების არსებობისას (გაფიცვა, ლოკაუტი, ანაზღაურების გარეშე შვებულება, სამხედრო სავალდებულო სამსახურში გაწვევა და ა.შ.) თუ დასაქმებულ პირზე მინიმუმ 3 საანგარიშო პერიოდის მანძილზე არ გაიცემა ხელფასი;

შეწყვეტილი - პირს შეწყვეტილი სტატუსი ენიჭება იმ შემთხვევაში თუ შეწყდა მასთან შრომითი ხელშეკრულება;

დასაქმებული/დაქირავებული - დასაქმებული არის ფიზიკური პირი, რომელიც შრომითი ხელშეკრულების საფუძველზე, დამსაქმებლისათვის ასრულებს გარკვეულ სამუშაოს;

## Eservice API

ყველა API წვდომა გადის HTTPS და ვებ-სერვისის მისამართია: <https://eapi.rs.ge>

ყველა მონაცემი იგზავნება და მიიღება JSON-ში. გაითვალისწინეთ JSON-ში მნიშვნელობა აქვს ზედა და ქვედა რეგისტრს (ანუ DATA და Data სხვადასხვა). ამიტომ გამოიყენეთ ზუსტად ისე, როგორც მითითებულია დოკუმენტში.

ჩვენი სერვისი იყენებს API გასაღებს, იმისთვის რომ ავტორიზებული გახადოს მოთხოვნა. თქვენ შეგიძლიათ ნახოთ თქვენი გასაღები ავტორიზაციის გავლის შემდეგ პასუხში.

თქვენ გასაღებს აქვს ბევრი პრივილეგია, ასე რომ დარწმუნებული უნდა იყოთ მის უსაფრთხოებაში. არ გააზიაროთ თქვენი გასაღები საჯაროდ ხელმისაწვდომ ადგილებში, მაგალითად როგორიცაა GitHub, კოდი კლიენტის მხარეს და ა.შ.

იმისთვის, რომ გამოიყენოთ გასაღები უნდა მიაწოდოთ თავში Authorization პარამეტრს ყველა მეთოდში:

"Authorization: bearer {ACCESS\_TOKEN}"

ყველა API მოთხოვნა უნდა განხორციელდეს HTTPS-ზე, მოთხოვნები გაკეთებული HTTP- დან და ასევე ყველა მოთხოვნა ავტორიზაციის გარეშე ვერ განხორციელდება.

ყველა გაგზავნილ მოთხოვნაზე ბრუნდება ერთი სტილის პასუხი შესაბამისი მონაცემებით და სტატუსით. შეიცავს ორ ობიექტს: DATA და STATUS. თუ მოთხოვნამ წარმატებით ჩაიარა DATA ობიექტი შევსებული დაბრუნდება თუ არა ცარიელი. DATA ობიექტის ველები განსხვავებულია თითოეული მეთოდისთვის, რასაც განვიხილავთ თითოეულის აღწერისას თავში ['საგადასახადო დოკუმენტი'](#). STATUS ობიექტი სტანდარტულია და შეიცავს ორ ველს: ID და TEXT. მისი ყველა შესაძლო მნიშვნელობა აღწერილია თავში ['შეცდომები'](#).

მაგალითად შეცდომის პასუხი:

### RESPONSE

```
{
  "DATA": {},
  "STATUS": {
    "ID": -1,
    "TEXT": "სისტემური შეცდომა. გთხოვთ სცადოთ მოგვიანებით."
  }
}
```

## შეცდომები

ვიყენებთ HTTP პასუხის კოდებს, რომ მივუთითოთ API მოთხოვნის წარმატებით ან წარუმატებლად დასრულების შესახებ. კოდები **2xx** დიაპაზონში მიუთითებს წარმატებას. კოდები **4xx** დიაპაზონში მიუთითებს შეცდომას, რომელიც ვერ მოიცავს მოცემულ ინფორმაციას. კოდები **5xx** დიაპაზონში მიუთითებს შეცდომებს სერვერის მხარეზე.

### HTTP სტატუს კოდები

200 - OK	ყველაფერი მუშაობს
400 - Bad Request	სავარაუდოდ აუცილებელი პარამეტრი არ გადაეცემა
401 - Unauthorized	არაავტორიზებული წვდომა
403	შესვლა აკრძალულია
404 - Not Found	მოთხოვნილი რესურსი არ არსებობს
500, 503 - Server Errors	სერვერის მხარეს რაღაც წესრიგში არ არის (მაგ. 503 სერვერი დროებით მიუწვდომელია)

შეცდომები, რომელსაც ვაბრუნებთ API მოთხოვნის წარმატებით ან წარუმატებლად დასრულების შემთხვევაში პასუხად არის შემდეგი: კოდები **-1xx** დიაპაზონში მიუთითებს ავტორიზაციის შეცდომებს, კოდები **-2xx** დიაპაზონში მიუთითებს საგადასახადო დოკუმენტის შეცდომებს.

### შეცდომები API exceptions [ ID – TEXT ]

0	ოპერაცია წარმატებით დასრულდა
-1	სისტემური შეცდომა. გთხოვთ სცადოთ მოგვიანებით.
-2	არასწორი ქმედება
-3	გთხოვთ გამოიყენოთ სატესტო კოდი : 206322102 ან 12345678910
...	...

ჩვენი API ბევრ მიზეზე აბრუნებს შეცდომებს, მაგალითად როგორცაა არასწორი პარამეტრები, ავტორიზაციის შეცდომა, კავშირის შეცდომა და ა.შ. ჩვენ გირჩევთ დაწეროთ ისეთი კოდი, რომელიც მოიცავს ყველა შეცდომას.

## ავტორიზაცია

[ POST: /Users/Authenticate ]

ავტორიზაცია გვაქვს ორი სახის:

1. ერთბიჯიანი - ავტორიზაცია ხორციელდება მომხმარებლით და პაროლით.
2. ორბიჯიანი - ავტორიზაცია ხორციელდება მომხმარებლით, პაროლით და ტელეფონზე მოსული ერთჯერადი კოდით.

ავტორიზაციის გავლა ხორციელდება Authenticate მეთოდით, იმის მიხედვით მომხმარებელს ჩართული აქვს თუ არა ორბიჯიანი ავტორიზაცია ბრუნდება სხვადასხვა პასუხები. თუ ორბიჯიანი ავტორიზაცია არ აქვს ჩართული (ან მოწყობილობის კოდი ემთხვევა შენახულს) ავტორიზებული გასაღები ბრუნდება, თუ ჩართული აქვს (და მოწყობილობის კოდი არ ემთხვევა შენახულს ან ცარიელი გადაეცემა) მაშინ დროებითი გასაღები ბრუნდება, რომელიც ტელეფონზე მოსულ კოდთან ერთად უნდა გადავცეთ AuthenticatePin მეთოდს, რომ მივიღოთ ავტორიზებული გასაღები.

პირველ ეტაპზე ვიძახებთ Authenticate მეთოდს:

REQUEST	პარამეტრები	/Users/Authenticate
USERNAME		
String [სავალდებულო]		მომხმარებლის სახელი ან პირადი ნომერი
PASSWORD		
String [სავალდებულო]		პაროლი
DEVICE_CODE		
String		ორბიჯიანი ავტორიზაციის შემთხვევაში დამახსოვრებული მოწყობილობის კოდი GUID.

იმის მიხედვით რომელ პასუხს დააბრუნებს (ერთბიჯიანისას თუ ორბიჯიანისას) პროცესი გრძელდება განსხვავებულად. თუ პირვლივე ბიჯზე დასრულდა მაშინ პასუხი შემდეგნაირია:

RESPONSE	პარამეტრები	DATA ობიექტში ერთბიჯიანი ავტორიზაციისას
ACCESS_TOKEN		
String		გასაღები
EXPIRES_IN		
Number		ვადის გასვლის დრო (წამებში)



MASKED\_MOBILE  
String

დაფარული ტელეფონის ნომერი (ბოლო ორი ციფრი ჩანს) გამოიყენება ორბიჯიანი ავტორიზაციისას კოდის მიმღები ტელეფონის გამოსატანად

თუ სჭირდება მეორე ეტაპზე გადასვლა მაშინ პასუხი შემდეგნაირია:

RESPONSE პარამეტრები DATA ობიექტში ორბიჯიანი ავტორიზაციისას

PIN\_TOKEN  
String

დროებითი გასაღები

MASKED\_MOBILE  
String

დაფარული ტელეფონის ნომერი (ბოლო ორი ციფრი ჩანს) გამოიყენება ორბიჯიანი ავტორიზაციისას კოდის მიმღები ტელეფონის გამოსატანად

თუ მეორე ეტაპის პასუხი დაგიბრუნდათ, მაშინ დროებითი გასაღებით და ტელეფონზე მოსული კოდით უნდა მიიღოთ გასაღები შემდეგი მეთოდის გამოძახებით:

REQUEST პარამეტრები /Users/AuthenticatePin

PIN\_TOKEN  
String [სავალდებულო]

დროებითი გასაღები, რომელიც გიბრუნდებათ Authenticate-ის გამოძახებისას

PIN  
String [სავალდებულო]

ტელეფონზე მოსული ოთხნიშნა ერთჯერადი კოდი

DEVICE\_CODE  
String

ორბიჯიანი ავტორიზაციის შემთხვევაში დასამახსოვრებელი მოწყობილობის კოდი GUID

ADDRESS  
String

მისამართი, საიდანაც ხდება სისტემაში შესვლა

BROWSER  
String

ბროუზერი, რომლის გამოყენებითაც ხდება სისტემაში შესვლა

OPER\_SYSTEM  
String

ოპერაციული სისტემა, რომელზეც გაშვებულია შესაბამისი პროგრამა თუ ბროუზერი

RESPONSE პარამეტრები DATA ობიექტში

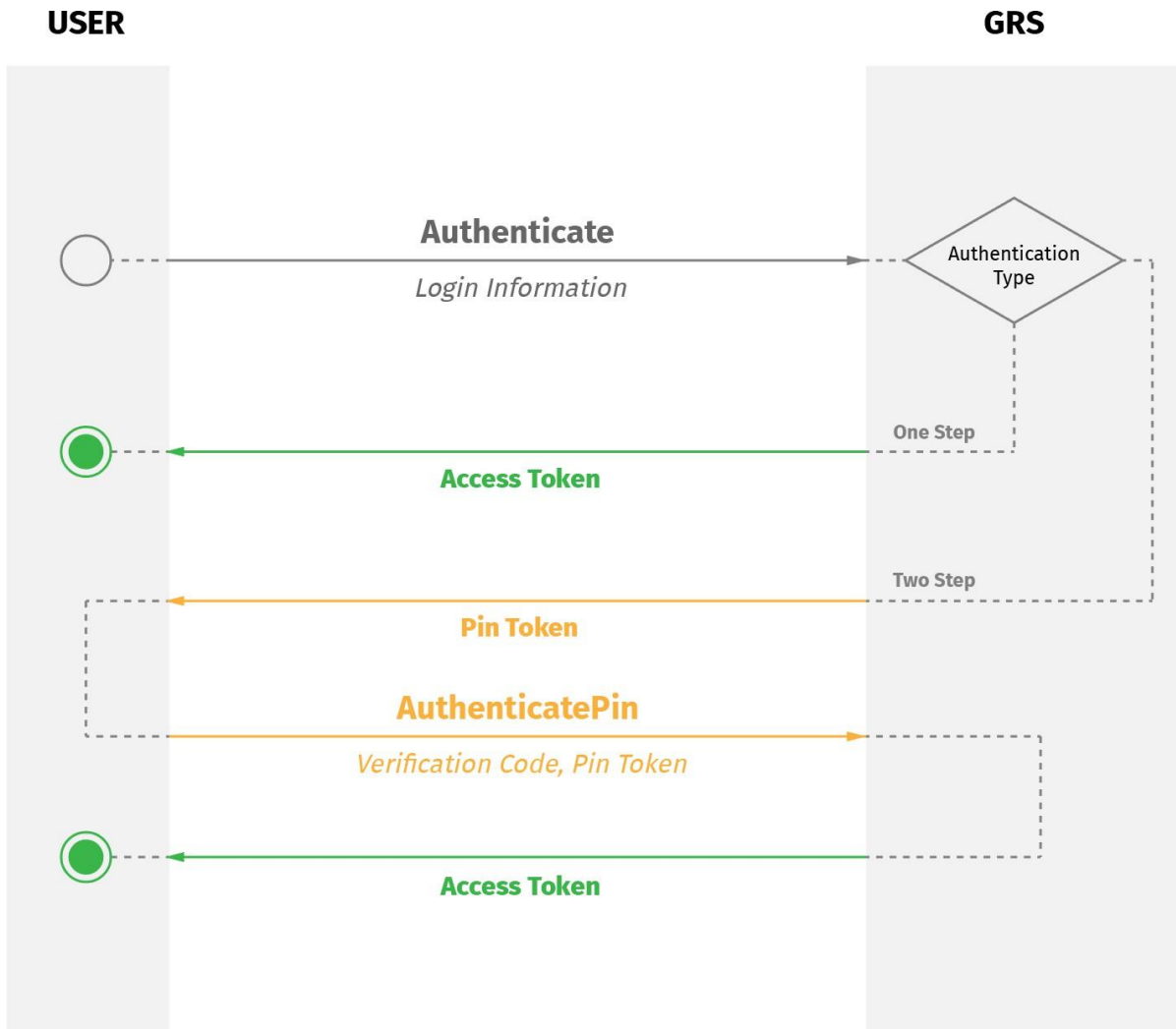
ACCESS\_TOKEN  
String

გასაღები

EXPIRES\_IN  
Number

ვადის გასვლის დრო (წამებში)

ავტორიზაციის სქემა:



{DEVICE\_CODE}, {ADDRESS}, {BROWSER}, {OPER\_SYSTEM} ივსება მხოლოდ იმ შემთხვევაში თუ გვსურს იმ მოწყობილობის დამახსოვრება საიდანაც შედის, რათა შემდეგ შესვლაზე მხოლოდ მომხმარებლით და პაროლით შეძლოს შესვლა. ამ შემთხვევაში მნიშვნელოვანია {DEVICE\_CODE}, რითიც ხდება გარკვევა მოწყობილობა დამახსოვრებულია თუ არა, დანარჩენი ველები ინფორმაციული ხასიათისაა და გამოიყენება მომხმარებლისთვის დამატებითი ინფორმაციის მიწოდების მიზნით.

შესაძლო შეცდომები:

- [-1](#)
- [-1xx](#)

თქვენ შეგიძლიათ სერვისი გაუშვათ სატესტო რეჟიმში, რომელიც არ შეცვლის ნამდვილ მონაცემებს. დასატესტად შეგიძლიათ გამოიყენოთ სატესტო მომხმარებელი.

მომხმარებლის სახელი	პაროლი	პირადი ნომერი
Tbilisi	123456	206322102
Satesto2	123456	12345678910

მაგალითი

განვიხილოთ მაგალითი სატესტო მომხმარებლის შემთხვევაში:

ერთბიჯიანი ავტორიზაცია:

POST /Users/Authenticate

Host: eapi.rs.ge

Content-Type: application/json

```
{
  "USERNAME" : "tbilisi",
  "PASSWORD" : 123456,
  "DEVICE_CODE" : null
}
```

RESPONSE

```
{
  "DATA": {
    "ACCESS_TOKEN": "190cf874-cb7c-437f-82c4-1a7725d40f4f-10042019032106",
    "EXPIRES_IN": 2400,
    "MASKED_MOBILE": "*** *** *16"
  },
  "STATUS": {
    "ID": 0,
    "TEXT": "ოპერაცია წარმატებით დასრულდა"
  }
}
```

ორბიჯიანი ავტორიზაცია:

პირველი ეტაპი:

POST /Users/Authenticate

Host: eapi.rs.ge

Content-Type: application/json

```
{
  "USERNAME": "satesto2",
  "PASSWORD": 123456,
  "DEVICE_CODE": null
}
```

RESPONSE

```
{
  "DATA": {
    "PIN_TOKEN": "roTFAvUvIyLFHDqYOKAp-10042019171434", "MASKED_MOBILE":
    "*** *** *28"
  }
}
```

```

},
"STATUS":{
  "ID": 0,
  "TEXT": "ოპერაცია წარმატებით დასრულდა"
}
}

```

მეორე ეტაპი (მოწყობილობის დამახსოვრების გარეშე):

POST /Users/AuthenticatePin

Host: eapi.rs.ge

Content-Type: application/json

```

{
  "PIN_TOKEN": "roTFAvUvIyLFHDqYOKAp-10042019171434",
  "PIN": 7542,
  "DEVICE_CODE": null,
  "ADDRESS": null, "BROWSER":
  null, "OPER_SYSTEM": null
}

```

ან როცა გვინდა მოწყობილობის დამახსოვრება:

```

{
  "PIN_TOKEN": "roTFAvUvIyLFHDqYOKAp-10042019171434",
  "PIN": 7542,
  "DEVICE_CODE": "df63cf1b-7417-49c1-87fb-e82037364746",
  "ADDRESS": "Tbilisi, Georgia",
  "BROWSER": "Chrome 73",
  "OPER_SYSTEM": "Windows 10"
}

```

RESPONSE

```

{
  "DATA": {
    "ACCESS_TOKEN": "ff68b0d6-7733-413f-a9d3-f2b3bed05d59-10042019052459",
    "EXPIRES_IN": 2400
  },
  "STATUS": {
    "ID": 0,
    "TEXT": "ოპერაცია წარმატებით დასრულდა"
  }
}

```

\*გაითვალისწინეთ, რომ ქვე-მომხმარებლით შესვლისას მონიშნული/ჩართული უნდა გქონდეთ საგადასახადო დოკუმენტის შესაბამისი უფლებები „მომხმარებლების მართვა“ მოდულში შესაბამისი მომხმარებლისთვის.

## სისტემიდან გამოსვლა

[ POST: /Users/SignOut ]

სისტემიდან გამოსასვლელად (გასაღების გასაუქმებლად) უნდა გამოიძახოთ SignOut მეთოდი, რომელსაც პარამეტრები არ გადაეცემა. თავში უნდა გადასცეთ გასაღები "Authorization: bearer {ACCESS\_TOKEN}" (sent in a header).

შესაძლო შეცდომები:

- [-1](#)
- [-104](#)

მაგალითი

განვიხილოთ მაგალითი სატესტო მომხმარებლის შემთხვევაში:

POST /Users/SignOut

Host: eapi.rs.ge

Authorization: bearer ff68b0d6-7733-413f-a9d3-f2b3bed05d59-10042019052459

Content-Type: application/json

## RESPONSE

```
{
  "DATA": {},
  "STATUS": {
    "ID": 0,
    "TEXT": "ოპერაცია წარმატებით დასრულდა"
  }
}
```

## დაქირავებულ პირთა რეესტრი

[ <https://eapi.rs.ge/Employees> ]

დაქირავებულ პირთა რეესტრის მეთოდების გამოსამახებლად მომხმარებელს უნდა ჰქონდეს ჩართული აღნიშნული მოდული. წინააღმდეგ შემთხვევაში დაბრუნდება შეცდომა. ასევე თითოეული მეთოდი სხვადასხვა ნებართვას მოითხოვს რისი არ ქონაც ნიშნავს ოპერაციის წარუმატებლად დასრულებას. ნებართვები/უფლებები გაწერილი აქვს თითოეულ მეთოდს სათაურის ქვეშ.

თარიღის ფორმატია: 'დღე-თვე-წელი დრო' - "DD-MM-YYYY 24HH:MI:SS".

### EMPLOYEE ობიექტი

EMPLOYEE	
ID Number [სავალდებულო]	დოკუმენტის საიდენტიფიკაციო ნომერი, შენახვისას გადაეცემა 0
IS_FOREIGNER Number [სავალდებულო]	0 - საქართველოს მოქალაქე, 1 - უცხო ქვეყნის მოქალაქე
CITIZEN_COUNTRY_ID String	უცხო ქვეყნის კოდი, ივსება როცა IS_FOREIGNER=1
STATUS Number [სავალდებულო]	სტატუსი: 1 - აქტიური, 0 - შეწყვეტილი, -1 - შეჩერებული
TIN String [სავალდებულო]	დაქირავებული პირის ს/ნ ან პ/ნ
FULLNAME String	დაქირავებული პირის სახელი, გვარი, ივსება როცა IS_FOREIGNER=1
PHONE String [სავალდებულო]	დაქირავებული პირის მობილური ტელეფონის ნომერი
GENDER Number	სქესი: 1 - მამრობითი, 2 - მდედრობითი. ივსება როცა IS_FOREIGNER=1
BIRTH_DATE String	დაქირავებული პირის დაბადების თარიღი, ივსება როცა IS_FOREIGNER=1
WORK_TYPE Number [სავალდებულო]	სამუშაოს ტიპი: 1 - სრული, 2 - ნახევარი

დაქირავებული პირის მონაცემები ID-ის მიხედვით

[ POST: /Employees/GetEmployee ]

[ნებართვა: წაკითხვა ]

GetEmployee მეთოდი აბრუნებს კონკრეტული ჩანაწერის მონაცემებს ID-ის მიხედვით.

REQUEST	პარამეტრები	/Employees/GetEmployee
ID		შექმნილი დოკუმენტის საიდენტიფიკაციო ნომერი
Number	[სავალდებულო]	
RESPONSE	პარამეტრები	DATA ობიექტში
EMPLOYEE		მონაცემების <a href="#">ობიექტი</a>
Object		

შესაძლო შეცდომები:

- [-1](#)
- [-30](#)
- [-104](#)

მაგალითი

განვიხილოთ მაგალითი სატესტო მომხმარებლის შემთხვევაში:

POST /Employees/GetEmployee
Host: eapi.rs.ge
Authorization: bearer ff68b0d6-7733-413f-a9d3-f2b3bed05d59-10042019052459
Content-Type: application/json
{
"ID": 2100
}
RESPONSE
{
"DATA": {
"EMPLOYEE": { "STATUS":
"-1",
"STATUS_TXT": "შეჩერებული", "IS_FOREIGNER":
"დიახ",
"CITIZENSHIP": "ავსტრალია", "TIN":
"22221111",
"FULLNAME": "dad dadaa", "PHONE":
"22222222",
"GENDER": "1",
"GENDER_TXT": "მამრობითი",
"BIRTH_DATE": "13-12-2021",
"WORK_TYPE": "1",
"WORK_TYPE_TXT": "სრული",
"IS_CORRECTED": "1",

```

        "IS_CORRECTED_TXT": "დიახ",
        "CITIZEN_COUNTRY_ID": "036",
        "CREATE_DATE": "13-12-2021", "ACTIVATE_DATE":
        "",
        "CANCEL_DATE": "",
        "SUSPEND_DATE": "13-12-2021"
    }
},
"STATUS":{
    "ID": 0,
    "TEXT": "ოპერაცია წარმატებით დასრულდა"
}
}

```

დაქირავებული პირის შენახვა

[ POST: /Employees/SaveEmployee ]

[ ნებართვა: შექმნა/რედაქტირება ]

SaveEmployee მეთოდი ინახავს/არედაქტირებს დაქირავებული პირის მონაცემებს. ID- ის შევსება სავალდებულოა. ახალი ჩანაწერის შენახვისას ID უნდა იყოს 0. თუ ახდენთ დოკუმენტის რედაქტირებას უნდა გადასცეთ იმ ჩანაწერი ID, რომელსაც არედაქტირებთ.

REQUEST	პარამეტრები	/Employees/SaveEmployee
EMPLOYEE		მონაცემების <a href="#">ობიექტი</a>
Object [სავალდებულო]		
RESPONSE	პარამეტრები	DATA ობიექტში
ID		შექმნილი დოკუმენტის საიდენტიფიკაციო ნომერი
Number		

შესაძლო შეცდომები:

- [-1](#)
- [-7](#)
- [-104](#)
- [-8xx](#)

მაგალითი

განვიხილოთ მაგალითი სატესტო მომხმარებლის შემთხვევაში:

POST /Employees/SaveEmployee

Host: eapi.rs.ge



Authorization: bearer ff68b0d6-7733-413f-a9d3-f2b3bed05d59-10042019052459

Content-Type: application/json

```
{
  "ID": 0,
  "IS_FOREIGNER": 1,
  "TIN": "12345678910",
  "FULLNAME": "dff dd",
  "GENDER": "1",
  "CITIZEN_COUNTRY_ID": "040",
  "PHONE": "500000004",
  "WORK_TYPE": 1,
  "BIRTH_DATE": "19-10-1990"
}
```

## RESPONSE

```
{
  "DATA": {
    "ID": "2115"
  }, "STATUS": {
    "ID": 0,
    "TEXT": "ოპერაცია წარმატებით დასრულდა"
  }
}
```

რედაქტირების შემთხვევაში გადაეცემა ID:

```
{
  "ID": 2115,
  "IS_FOREIGNER": 1,
  "TIN": "12345678910",
  "FULLNAME": "dff eee dd", "GENDER": "1",
  "CITIZEN_COUNTRY_ID": "040",
  "PHONE": "500000004",
  "WORK_TYPE": 1,
  "BIRTH_DATE": "19-10-1990"
}
```

დაქირავებულ პირთა სია

[ POST: /Employees/ListEmployees ]

[ნებართვა: წაკითხვა ]

ListEmployees მეთოდი აბრუნებს დაქირავებულ პირთა სიას. გადაეცემა ფილტრის პარამეტრები, რომლებიც არასავალდებულოა.

REQUEST	ფილტრის პარამეტრები	/Employees/ListEmployees
ID		საიდენტიფიკაციო ნომერი
Number		
TIN		დაქირავებული პირის პირადი ნომერი
String		
FULLNAME		დაქირავებული პირის სახელი, გვარი
String		
GENDER		სქესი: 1 - მამრობითი, 2 - მდედრობითი
Number		
CITIZENSHIP		მოქალაქეობა (რამდენიმეს გადაცემის შემთხვევაში გამოყავით მძიმით მაგ.: საქართველო,ავსტრიის რესპუბლიკა,...)
String		
STATUS		სტატუსი: 1 - აქტიური, 0 - გაუქმებული, -1 - შეჩერებული
Number		
PHONE		მობილური ტელეფონის ნომერი
String		
WORK_TYPE		სამუშაოს ტიპი: 1 - სრული, 2 - ნახევარი
Number		
BIRTH_DATE		დაბადების თარიღი (დიაპაზონი „საწყისი თარიღი : საბოლოო თარიღი“, ფორმატი “DD-MM-YYYY:DD-MM-YYYY”)
String		
CREATE_DATE		შექმნის თარიღი (დიაპაზონი „საწყისი თარიღი : საბოლოო თარიღი“, ფორმატი “DD-MM-YYYY:DD-MM- YYYY”)
String		
ACTIVATE_DATE		გააქტიურების თარიღი (დიაპაზონი „საწყისი თარიღი : საბოლოო თარიღი“, ფორმატი “DD-MM-YYYY:DD-MM-YYYY”)
String		
CANCEL_DATE		შეწყვეტის თარიღი (დიაპაზონი „საწყისი თარიღი : საბოლოო თარიღი“, ფორმატი “DD-MM-YYYY:DD-MM-YYYY”) გადაეცემა იმ შემთხვევაში თუ status = 0
String		
SUSPEND_DATE		შეჩერების თარიღი (დიაპაზონი „საწყისი თარიღი : საბოლოო თარიღი“, ფორმატი “DD-MM-YYYY:DD-MM-YYYY”) გადაეცემა იმ შემთხვევაში status = -1
String		
MAXIMUM_ROWS		დაქირავებული პირების რაოდენობა
Number		

RESPONSE პარამეტრები DATA ობიექტში არის Data და Summary ობიექტები. Summary შეიცავს ერთ ველს \_Count, რომელშიც ბრუნდება ჩანაწერების რაოდენობა. Data ობიექტი შეიცავს ორ ობიექტს: Fields და Rows. Fields შეიცავს ველების სახელების სიას, Rows შეიცავს ამ ველების შესაბამისი მონაცემების სიას.

#### RESPONSE პარამეტრები DATA.Data.Fields ობიექტში

ID	დოკუმენტის საიდენტიფიკაციო ნომერი
Number	
TIN	დაქირავებული პირის ს/ნ ან პ/ნ
String	
ORG_TIN	დამქირავებლის ს/ნ ან პ/ნ
String	
ORG_FULLNAME	დამქირავებლის დასახელება
String	
CITIZENSHIP	მოქალაქეობა
String	
FULLNAME	დაქირავებული პირის სახელი, გვარი
String	
PHONE	მობილური ტელეფონის ნომერი
String	
GENDER	სქესი: 1 - მამრობითი, 2 - მდედრობითი
Number	
GENDER_TXT	სქესი: მამრობითი, მდედრობითი
String	
BIRTH_DATE	დაბადების თარიღი
String	
CREATE_DATE	შექმნის თარიღი
String	
ACTIVATE_DATE	გააქტიურების თარიღი
String	
CANCEL_DATE	შეწყვეტის თარიღი
String	
WORK_TYPE	სამუშაოს ტიპი
Number	
WORK_TYPE_TXT	სამუშაოს ტიპი
String	
STATUS	სტატუსი
Number	
STATUS_TXT	სტატუსი
String	

IS_FOREIGNER	უცხო ქვეყნის მოქალაქე
Number	
IS_CORRECTED	რედაქტირებულია: 1 - დიახ, 0 - არა
Number	
IS_CORRECTED_TXT	რედაქტირებულია: დიახ/არა
String	
CITIZEN_COUNTRY_ID	უცხო ქვეყნის ID
String	
LAST_CHANGE_DATE	ბოლო ცვლილების თარიღი
String	
SUSPEND_DATE	შეჩერების თარიღი
String	

შესაძლო შეცდომები:

- [-1](#)
- [-104](#)

მაგალითი

განვიხილოთ მაგალითი სატესტო მომხმარებლის შემთხვევაში:

თუ ფილტრს არ გადასცემთ დაგიბრუნდებათ ბოლო 10 ბაზაში არსებული დოკუმენტი, თუ

ფილტრს გადასცემთ დაგიბრუნდებათ შესაბამისი დოკუმენტი არსებობის შემთხვევაში (დიაპაზონში თარიღები უნდა გამოყოფილი იყოს ':' სიმბოლოთი)

POST /Employees/ListEmployees

Host: eapi.rs.ge

Authorization: bearer ff68b0d6-7733-413f-a9d3-f2b3bed05d59-10042019052459

Content-Type: application/json

```
{
  "STATUS": "-1",
  "TIN": "22221111"
}
```

RESPONSE

```
{
  "DATA": {
    "Data": {
      "Fields": [
        "ID",
        "TIN", "ORG_TIN",
        "ORG_FULLNAME",
        "CITIZENSHIP",
        "FULLNAME",
        "PHONE",
        "GENDER",
        "GENDER_TXT",
        "BIRTH_DATE",
        "CREATE_DATE",
        "ACTIVATE_DATE",
        "CANCEL_DATE",
        "WORK_TYPE",

```

```

        "WORK_TYPE_TXT",
        "STATUS", "STATUS_TXT",
        "IS_FOREIGNER",
        "IS_CORRECTED",
        "IS_CORRECTED_ICON",
        "IS_CORRECTED_TXT",
        "CITIZEN_COUNTRY_ID",
        "LAST_CHANGE_DATE",
        "SUSPEND_DATE"
    ],
    "Rows": [ [
        2100.0,
        "22221111",
        "206322102",
        "სატესტო კოდი1",
        "ავსტრალია", "dad
        dadaa", "22222222",
        1.0,
        "მამრობითი",
        "13-12-2021",
        "13-12-2021",
        null,
        null,
        1.0,
        "სრული",
        -1.0,
        "შეჩერებული", 1.0,
        1.0,
        "done",
        "დიახ",
        "036",
        "13-12-2021",
        "13-12-2021"
    ]
    ],
    },
    "Summary": {
        "_Count": 1.0
    }
    },
    "STATUS": {
        "ID": 0,
        "TEXT": "ოპერაცია წარმატებით დასრულდა"
    }
    }
}

```

## ქვეყნების ჩამონათვალი

[ POST: /Employees/GetCountries ] [

ნებართვა: წაკითხვა ]

GetCountries მეთოდი აბრუნებს ქვეყნების ჩამონათვალს. ფილტრის გადაცემის შემთხვევაში დაბრუნდება შესაბამისი მონაცემები, ფილტრის გარეშე დაბრუნდება ყველა ქვეყანა.

REQUEST პარამეტრები /Employees/GetCountries

COUNTRY\_NAME  
String ქვეყნის დასახელება

COUNTRY\_ID  
String ქვეყნის კოდი

RESPONSE პარამეტრები DATA ობიექტში

COUNTRIES  
List<Object> ქვეყნები

შესაძლო შეცდომები:

□ [-1, -104](#)

მაგალითი

განვიხილოთ მაგალითი სატესტო მომხმარებლის შემთხვევაში:

POST /Employees/GetCountries

Host: eapi.rs.ge

Authorization: bearer ff68b0d6-7733-413f-a9d3-f2b3bed05d59-10042019052459

Content-Type: application/json

```
{
  "COUNTRY_NAME": "რეს",
  "COUNTRY_ID": "03"
}
```

RESPONSE

```
{
  "DATA": {
    "COUNTRIES": [
      {
        "country_id": "031",
        "name": "აზერბაიჯანის რესპუბლიკა"
      }
    ]
  },
  "STATUS": {
    "ID": 0,
    "TEXT": "ოპერაცია წარმატებით დასრულდა"
  }
}
```