



Face-Intellect. REST API

## Table of contents

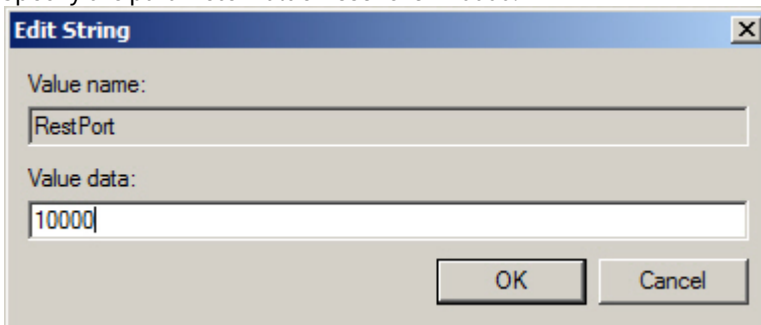
<b>1</b>	<b>General information.....</b>	<b>4</b>
<b>2</b>	<b>Getting the recognition protocol .....</b>	<b>5</b>
2.1	General request format:.....	5
2.2	Request parameters:.....	5
2.3	Request example:.....	5
2.4	Response example: .....	6
2.5	Response parameters: .....	7
<b>3</b>	<b>Adding/updating user in face database.....</b>	<b>9</b>
3.1	General request format:.....	9
3.2	Request parameters:.....	9
3.3	Request example:.....	10
3.4	Response example: .....	12
3.5	Response parameters: .....	13
<b>4</b>	<b>Updating user in face database.....</b>	<b>14</b>
4.1	General request format:.....	14
4.2	Request parameters:.....	14
4.3	Request example:.....	14
4.4	Response example: .....	15
4.5	Response parameters: .....	15
<b>5</b>	<b>Getting information about users.....</b>	<b>16</b>
5.1	General request format:.....	16
5.2	Request parameters:.....	16
5.3	Request example:.....	16
5.4	Response example: .....	16
5.5	Response parameters: .....	17
<b>6</b>	<b>Deleting users from the database .....</b>	<b>18</b>
6.1	General request format:.....	18
6.2	Request parameters:.....	18
6.3	Request example:.....	18
6.4	Response example: .....	18
6.5	Response parameters: .....	18

<b>7</b>	<b>Searching for faces by image</b>	<b>20</b>
7.1	General request format:	20
7.2	Request parameters:	20
7.3	Request example:	20
7.4	Response example:	22
7.5	Response parameters:	23
<b>8</b>	<b>Getting image by ID</b>	<b>25</b>
8.1	General request format:	25
8.2	Request parameters:	25
8.3	Request example:	25
8.4	Response example:	25
<b>9</b>	<b>Getting analytic data</b>	<b>26</b>
9.1	General request format:	26
9.2	Request parameters:	26
9.3	Request example:	26
9.4	Response example:	27
9.5	Response parameters:	28
<b>10</b>	<b>Getting analytic data by time</b>	<b>30</b>
10.1	General request format:	30
10.2	Request parameters:	30
10.3	Request example:	30
10.4	Response example:	31
10.5	Response parameters:	32
<b>11</b>	<b>List of possible errors of the Face Recognition Server</b>	<b>33</b>
11.1	General request format:	33
11.2	Request example:	33
11.3	Response:	33
11.4	Response parameters:	33

# 1 General information

Before start of using the REST API and launch the *Face-Intellect* software package, do the following:

1. Select the HKEY\_LOCAL\_MACHINE\SOFTWARE\Wow6432Node\ITV\INTELLECT\FRS registry section.
2. Specify the parameter value **RestPort** = 10000.



## 2 Getting the recognition protocol

### 2.1 General request format:

POST <http://IP-address:port/firserver/GetProtocols>

### 2.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	ID of the Face Recognition Server in the <i>Face-Intellect</i> software package
onlineRefresh	Yes	This parameter must be set to 1
genders	No	Gender: 0 - undefined, 1 - male, 2 - female If not specified, gender is not taken into account
page	No	Number of results pages If not specified, then page = 1
pageSize	No	Results per page If not specified, then pageSize = 1, i.e. there will be 1 result on 1 page
cameraIds	No	Camera IDs (GUID or ID in the <i>Intellect</i> object tree) If not specified, then results for all cameras are displayed
sim_min	No	Face types: -1 - all faces, 1 - only recognized, 0 - only unrecognized If not specified, then sim_min = -1
minAge	No	Minimum age of persons If not specified, age is not taken into account
maxAge	No	Maximum age of persons If not specified, age is not taken into account
dateTimeFrom	Yes	The initial boundary of the time period in UTC+0 format: <b>yyyy-mm-ddThh:nn:ss.fff</b>
dateTimeTo	Yes	Ending boundary of the time period in UTC+0 format: <b>yyyy-mm-ddThh:nn:ss.fff</b>

### 2.3 Request example:

POST <http://127.0.0.1:10000/firserver/GetProtocols>

```
{
  "server_id": "1",
```

```

"onlineRefresh": 1,
"genders": [
  0,
  1,
  2
],
"dateTimeFrom": "2020-05-14T16:25:55.687",
"dateTimeTo": "2020-05-16T16:26:55.687",
"cameraIds": [
  "1"
],
"minAge": 0,
"maxAge": 1000,
"page": 1,
"pageSize": 50,
"sim_min": -1
}

```

## 2.4 Response example:

```

{
  "Protocols": [
    {
      "id": "18D10CC2-9596-EA11-8A9A-B06EBF8119EF",
      "camera_id": "5",
      "camera_name": "grundig 5",
      "timestamp": "2020-05-15T10:20:54.657",
      "age": 55.4815,
      "gender": 2,
      "emotion": 4,
      "glasses": 2,
      "facialHair": 6,
      "hairColor": 1,
      "hairType": 3,
      "headwear": 13,
      "liveness": 1,
      "evasion": 1,
      "temperature": 36.6,
      "fir": {
        "data": "rv0GvR00xbx76+a8kIHXPfbzkbwY7hA9vpS5PRdb0j32pg4+yqerPT6Ghjjw/
Vhy9PpMQvWHqmb2U5VQ8kic/vWIS/
b0xXNq8kJ7hPU7MGT2qdxw9FdAJPFxwTLxGLUQ9wBJCPLPiB71UMoU8vhbHP0Thqb3VaSg92qH0PHOj1zyEdaa9u10xvVfw
q71oy0o8twbWu3g+mD0SQvw8C/
KhvEjElrspZF19Q7xsu6FLol1mLa492n3tvCyBl7wNLl8+n+6RvRQnCbzDDhw9EnCAPdSsmT2PsHa8n+6HvTd6Cj0qT3U9o
FQMveIDNj0m7sU8eih0vYGvgT3INic9tWfXvevQnr1QMmG92WGHvf7Ulxz74a8aCmLvXAa6j0vEM08ApXTvGsowLw0uzC9
TpGt00eUfbzBOF4+SUGqPQpEvbyr+K08NJQTve1zZLrt6kQ9lvwEvZCE/
j0R9B693z+APJK7l7x6K+W9yPTm08ai1jp630U90KUUPujXET0WnPa8DIW7PE32Z73Fkke8kYNmvZg/
Lb2bv+o95BUKvYRH47yr70S8ojApvaMX5bwC4Co9AuwNvSFoWTyaXa09ueWFvPZ+RT3w7Mk7HnsLvDsmBD32PAW9IuphvSZ
LpTyBN5W9+e4sPcuAXr2RNI288I5hvYAm3LteHnm9FFVfIo8VL0ishw+QHjDPGGbhr2sJZK8E3a5PYsDK73kgKu8cN4/
PB3Z8z10AE+9HWm7PRJhdr5l7Tm9LjRHvZtTA7wTW+08nGa4PXmX0bs1mwk8xrGcPBgNfzyFF7A6vYPWPaRxUL3jK8e7Y3K
OvX5i77z543o9Xh/9PF0eAD24hrG8VYl1vRQnhrxGpcM8uu0C0dmlAz0z2AG+0cCHu6l/
Yz1Xdla9fHPnvAZuij2dv9g7h1bYu47LJb1fUFM9IpIcvSRgBr7RC4Y9GgFCPeuYTLyw2oe9wUZNPZAPGr0G1EK96x2nvY2
9ybyQdZE9/
v4jvcPhXbu9D309YwKWPYfTTD4S3fc9BtZqu3cFmLydFbu9Igsj01Pymjx0rUA9Z1qgvSmWSj3BB1m8DcuvPWBhR0e0YA9
r5eIPMbslL26mIi852wHPvVfOD1wWPK8gHECvSVsgr3bM7i9GN1eu+Lbfj0UDY29VU0hvv3xUzxaIAY9DtccPfk+BD0M+iy

```

```

9IZABvRr35DzOpsy9G7VWPVmkoD26GAE9JBptvbtnL73QTFA9+fs3veiXx7v76o09m130040zRT3zog09YTimvAln071mT
c8BPmCPcVQ8LzJ/yY9EdYUPSJpc704YWA90CaRvRmzOr6bnjg7vLHWPBcI0ryxdKq8WsZ9PIINOD00Rz09MVeIuw=="
    "engine": 3,
    "alg": 0
  },
  "Hits": [
    {
      "id": "A5204509-9596-EA11-8A9A-B06EBF8119EF",
      "person_id": "2",
      "person_guid": "D5E68CD2-F4D7-EB11-AC0A-B42E999B6711",
      "sim": 0.249181,
      "timestamp": "2020-05-15T10:15:44.807",
      "surname": "Doe",
      "name": "John",
      "patronymic": "White",
      "comment": "Senior developer",
      "department": "IT department"
    }
  ]
},
"ProtocolsTotal": 1,
>Status": "OK"
}

```

## 2.5 Response parameters:

Parameter	Parameter description
id	Identifier of the captured image
camera_id	Identifier of the camera that captured the face
camera_name	Name of the camera that captured the face;
timestamp	Face capture time in UTC+0 format
age	Age of face
gender	Gender: 0 - undefined, 1 - male, 2 - female
emotion	Emotion: 0 - undefined, 1 - rage, 2 - fear, 3 - happiness, 4 - without emotion, 5 - sadness, 6 - astonishment
glasses	Glasses: 0 - undefined, 1 - tinted glasses, 2 - without glasses, 3 - glasses
facialHair	Facial hair: 0 - undefined, 1 - beard, 2 - bristle, 3 - imperial beard and mustache, 4 - imperial beard, 5 - mustache, 6 - absent
hairColor	Hair color: 0 - undefined, 1 - brunet, 2 - blonde, 3 - chestnut hair, 4 - ginger, 5 - gray
hairType	Type of bald head: 0 - undefined, 1 - bald, 2 - with receding hairline, 3 - without bald head
headwear	Headwear: 0 - undefined, 1 - cap, 2 - bandana, 3 - beret, 4 - beanie, 5 - trapper hat, 6 - fur hood, 7 - glasses on head, 8 - hat, 9 - helmet, 10 - hood, 11 - kepi, 12 - headscarf, 13 - without headwear, 14 - peaky cap

Parameter	Parameter description
liveness	Artificial face: 0 - unknown, 1 - no, 2 - yes
evasion	Face concealment: 0 - unknown, 1 - face not covered, 2 - mask is on, 3 - tinted glasses, 4 - face partially covered, 5 - balaclava, 6 - other mask, 7 - mask is not on the right place
temperature	Temperature - face temperature in degrees Celsius
face_vector	Face vector data
face_data	Face pattern (vector)
engine	The used SDK: 1 - Cognitec, 2 - VISIONLABS, 3 - TEVIAN, 4 - STC
alg	The used face recognition and search algorithm: 0 - ALG1, 1 - ALG2, 2 - PRECISE
Hits	The recognized face information from the faceDB and intellect databases. If the face is not recognized, the field will be empty
id	The identifier of the image from the reference face database
person_id	The identifier of the corresponding user in <i>Intellect</i>
sim	Similarity rate of the captured face with the image from the reference face database
surname	Last name
name	First name
patronymic	Patronymic name
comment	Comment
department	Department
Status	Completed request status: OK - successful, if error - see <a href="#">List of possible errors of the Face Recognition Server</a>
person_guid	Identifier of the corresponding <i>Intellect</i> user in the GUID format

## 3 Adding/updating user in face database

✔ Also see [Updating user in face database](#)

### 3.1 General request format:

POST http://IP-address:port/firserver/CreatePerson

### 3.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	ID of Face Recognition Server in the face-Intellect software package
objectType	No	Object type
id	No	ID of the user in <a href="#">GUID</a> format If the ID is not specified, then it will be automatically generated for the newly added user and returned in the response;
name	No	Name If not specified, the person will be added without a name
surname	Yes	Surname
patronymic	No	Patronymic If not specified, the person will be added without a patronymic
department	Yes	Department
comment	No	Comment If not specified, the person will be added without a comment
createPersonInIntellect	No	The flag for creating a user in Intellect. If the parameter is equal to 1, the user will be created in Intellect on behalf of the <i>Intellect</i> operator If not specified, the <i>Intellect</i> user will not be created
image	Yes	JPEG image in the base64 format
fir	No	Face pattern (vector) If specified, the face will be added to the database without any verification (face search, face vector generation)



FjSPEKz3uka7JJp1wFkgR3RZIX+Zp4ZYVmkI2kqA5ZS+D8nGD1FL4we5ggaKcGRjIqMhwV2LMsFv7s5xgbcdTmvzc8JXx0u6ht9P0+IxXCST3Mm  
meapYnDR77fdNkku+zEibfmgGDcfanwoKa2unwvDdJdPhHzW9tJEXeVbUxGfFnZ503ZyYpKk3bm/u89EYYiTSqRXI/i0tp7vX5/  
wBXR7UJ4ZyYI5c3S7vr/n+P3eh674otLVYHmJPTk4nPMKSOXqJEF3v10zvb4yWzk/  
3a+cvFP7QV74UnuIr0G05LYPEN16FMJG6QKv25Y87idoIzsA9Mef/G1bjwzPivRwSOK07hmUTSIN0Uw8sLCiFskszkF8j0B8pGK/  
MrxRc3FrPcTeIJDxbD6kxjW5cosyrynsRht/mYYHGw7cbs1x1ea0Ip0qe03azV3sn/maVakKV0dSpFyhFXkLo2r20fc+o/C3xa8baxqYvb/  
xNdvFL5j20nWduYRCrn9+sK63LebnEAQ+Sm3a3yndgeGJ4501JIJNWTUPJJLHrhLhXCYLEyOpPmWMLgzuerZwASOp+ENL+MXGjRoVsrp9YtNcs  
1ZDDa2rSQgT8RyJOGQz58r0AibAByQ3FS6+Icd6kC0N2L60d5Jb37e8sdxBRkp5eyHcCruPM3IG+by1Hymtvqlep7jSSLo21ovh1f9fmrFKY7PMH  
G9J0qnJKydpNS7a3tp/Xy+q/GF5EwtruxRpBYaRcWdLNI6bjGXL30wCtLmYy24KE/  
L5XLVvuG38Qvj94Zi0P4j6ibScaQ216P7cL+SqLbqaI7p447cgeX9nNwu5xM3m+adsTan36zeG2fWNNt7vSoZTp98rxR3BjLR5Nm0XKpbyPJ5KfMn  
lv5h8wM/A2c/G37b3hb7HfDbWLe2Wyp31vSGtkImKyPDospupJ8RcuYiBAYGRuY+d8vPiSyicmfh6iqxbhUT5U/iaS6J637Ld/  
iYfHRrOpUqNGXsZx5ZSkM3y2WvNb5eVuh8C6QgkCR7dqAkSTE52FD8uExzuyx++u3HF0R9pfs7aaJ/  
FWjm6HnaZJHESsnGJPMtbs4aLZFLjKXbcAV2nbnduyPjPw/KI7mHzE3pCtQrJDnaRHNNcxGTftb/  
AFfk527fn8wDK7SW+9v2f9J1JPHFw+i0m1mvtW1DwbnS9F0e0M+ftfedAZTXPziF08xGNBLv8AmG4da+jwmGqyXNFVIvkc1zW0dt0v9fij5GLU  
lQxWLV02jktvfbLPV/8AhLN0/wCfC9/8B2/wor9Tf+GgvjF/0Lun/wDgvj/+SKK+o+oYftP/AMC/r+n6GX9o4nvD/  
wAAj5f5f1pb+YbxH42kutQldSiO44o4kjZDJkLxJKz7VYXGwdbnybPvNnitpl7cXa3U3mlwtlcmQdC0hChEBz1bnnbkYzgz1z7ofmXcwZmVwSyAp  
kS4+9L967NvGThs59ufcft4Pt737Abq184SzkRRYlsJDKmJNtP+tdjBYDbs0A3b57B4enUhgPjPm30dlftP8z1qNecqsIvls3ba39b1pb7P/  
AGZ/gzqur+E/DVwbaK031Gse4utSnGIIiWQRDOxYGMx5bY/zebubKIUBb7rvPgXdajZy6Po13dGeKEs1xb25e0nLL8v2Z/  
tC+Zt2uHwFC7l6545X4CDTNF8J6do83m1RLmSXtbcnNttm2+bDcwf8tli2ReUFmQjJf8v0K/Q3wbp9zFDBeSLHZM0YWRIGSW8gIwxit8J9mKgL/  
eSmXcvK+X82VevPDlocig0mnacVJP4d1p/SR9dhYueGnFbyikvmo/  
kfj7rHwE1PRjq0+ueI9e0m4CS2drBaieK09FwxWea4kjlksvLiKRmMlvtDgSjg7vSPhB8JfC/hi1m18QazP4/  
intoIxY7rizktJLFWY5Z7iU3uDuMubFEJ8wxN8yBRX60Xngvw5d7bvWUa+kAn2RRwsaxymbS0xfz8bCq7Plydz9Mccc3w+svMhTTNDsPsrPMbhgs  
NtIqkp507ZE3nFJSJSOV2H0PvVccwrSai4UffS6pJPPs76PT8WYYTKYUa/OLUXtH795N6aPTTT90U+Y/  
H4T0Xwvq0V7DgkkrzLyrhB0o1kwyYwVmmPs4LAMAPN3fZ29BAHPqVn4k+1aYtVbLHCrQ6WYhgjYqNmKkYfEwWqMqZf9/  
IUQqEj5v75otldTyhJBM4V/LMRcIYtxwjecndbyDwvQcHIpEtvS08EsBxI0rxKcZ+VSm5+G/2hx9ec4rb285e6+WzsvhXl/L/  
Wlvdo4KjGpBrnunp7z/rp/  
WhwHx2itdf0G0imjYy6ewe4hWQ4E5MJDN5FH7wxsdul2feb0R8veIvCwJ+MobWDWdFhvitoy2uB5ckTRLGAyzKjY3fJn5SSEHXfFYPjnS2uLS7g  
jUSiRyPLm9J27CocoiwZbcTlGf3q4440ePMNHsII5gEGcGiYxyggF92QWzgLcMCDZz160fHqz9njqE3ooyTu/  
RL+u53Y8PGVKCYpttalFqunU+M7D4Ig6lBdQ+FYPtVrLlLtc3WrK0ciEqAstsdOy+wIo+aXjfyxcV67p37PNjqN5bX3izw3YNGjLLHBpcwgWZlW5  
blliYbYjs2qU040/3cc/  
U2n+HtPjAdoLds28Fz5m052D5Q0ozzk8HARqhHMKRLHKmza4KrfTxb38xsjrjeh45ruxGN9pQqPpySnKNo23v7u1tFL5LuFPVMkoVpXrUfd0vZ  
uNkuXZ/1+J4Te+GdE0yykTTRe0UWlq++FLV47kRBFt4oVcXCb0BkEkg8y0b0wBRu/  
Mr9uSxiTwp0R8kd9Lp2qpfmKN+UMiJFJAGw2z0Qxk2n0wAxjg9v/FrWZSSKR4zMO0LUGHzd6x9D97ECZF2hvmMTnDdACrY4/  
Jz29s3F5p3hrS90wY3tZ1dNUT44NgW5bSPsr+28bclXyrjaMKH84gGPHzfHNUjKwZyao5txjVT1V09Laq+/m/  
kdzoUcHg61KjTiouFrtJyV2lpLdPb7j8xrTSrya+ha1C7ZNPijUk48qVxd3IlnfwfbCpThy5zu3DG2v3F/  
4Jk+DNN1TxxpXiTxPaNqs+l+Cn1jQLWkCwrafq0bXic9F15dyZGRoInCeRETkrvAGT+Ndspgv7mNyMLZV7qCMK+0Mb0Ly00NreFafPtR03nyxBj95  
5mV/cD/gmXPawEmpPPLP8AaoFAH2e0McPnEvEdVluMguPLI8+FzcZbRu44fFp0VzeIqUfaw965vZw6J6H5hmsVhKeKxFS04uT5vej8nT5n6z/  
wDC2fGv/P8A6j/4mF8A7nor52/ti99NQ/8AAMf/AB6ivrPq1LTL7/6/p+lvhf7Vxfeh/gC/z/q/  
pb+YfXvDTxNdeWoMqWd1cRnZsErQeV+4L7j5ZkEnD/Pt2n5Dk4+0vgT8MLPWT08IQ+Fr9Q01y/  
027v760ZGitzJ4xbRNLvMqLJakuN5TMxmHyR7Pm+ZPGXmwT7dW9xIIo4b97WMMdk4BEKbyfkE5LZBVthQD58/L9a/  
skftC+FPA9i3wr+IHIDTNE1C/  
wBc1G40TU5rCS+1gWwSWX7PY20ZjZ44P9HW5ka9iDCeBtgCEH4XL69BYfGSacvq8HKmk9XZxXzuv6Z+sYeLgvbU9JfEvtvtHy82fzfgHRJ9GmsDq  
UE1perXU8E2nT25UQmBoxmG5L4kvT2M+RGRtXAYH5fuzwvc+c1rEZDHPF8rFvMEJic8Kp2537wTym00tfJWt+MvBNudC1C/  
wDFWmXmt+Itk1naR3LmZplLeeZorIm1FyZLaaQRu0i0pdjhc5Jb1zjwr8waXzjtEbLFCpmQuvD7mwm4gY2fKSM8HkdjgrYnDV3f2NRVpSycnyp6b  
r5H22BhhY0nKnz0V4ktybV1y9PK3y/E+s9KmkUy7m2t5oi2F2o8QkM/  
l7+EbevLbc88Pu3KeAvII4CIPPVobTwnUwpJw+DGMn5R+AweDjka58y0vXTYfZbhLj95H5xJHR/  
M8vGefL2bSP4s7vb19/4sJELuXc3bsxhccgZxvJABxjKfXnpznKn8cfU7I+9JRdrPTt2/  
yX9W0k8ReIdK0QR5GwW5x5Swb9jKXwMmTD7j9EHpx25LQobi61C1Jc3YMc0qWca5um8zacW8QJ8/btG8/  
Iv+Tj5uPpp7051W6893lvWnbzI1htef3XLfQw8z5c71AJ54PAwK4PxF8Tvg3HDVY0PDYXGhDqL380L3NjmJ7BU2hZPtRtpPs4m30duxvME0dy7Mnu  
W69V+Z0KlCLTV7rzPetWsp5lla9iubeC1ac3VvLgyXdxKJt82MH90yBG8pMyGQsFubBu+evE922k3FtqnmQXKaPdSXCQSS7kllke3aDzfMt8EwKhn  
Ty2L0JgzckU5ta9+0f4i0Xsof+e0j7XcrHLJHJZPLFDfzykr7xf3kdvcFhbHy9im2G/  
wa2T7nr5fX4hfFDx14rN5NbrJowrn5LWIF4rGK1PCnknxDdIs/zufTeYwSvy8erjEpYimns2k+nSjvTV6kE9nJH2voerG/  
0+1uVkeGf0V3Dcn3eD3yc8ZaxjvXsX8cVuzSMFYidoLZyMHdzj0ACD0PuK8A0aa7srcIrtFggMvJAZcg4GQBzngZ6fnoajrcxiUPChCrJk5+7kL  
2y0dJ8Z46V7TowhOMLe6tdtrotV8v60tnXk1WhR05J6S0z1ZPFp8uxn+MdUIjyvtLgW/LoLmYN5fnAioTd/  
FHg4PX24HGa+CPijPZatG91qMsLWmtv000rXarbWcE0oUM1wMnLGMogj05vNy3+r2Yb6I8deLyLS9tLSV3d4ZEm2R+dvDzW8IjKjG3cJH0cn0za  
Aeo/Kr46/EgXPf8Q+FIDQuLzwl4d05NMu9KikvZ9QvIVkkumhTzDILHyhv3Db5wJL68AcFV3xlnWVnU6Ljy/meHmLT2MjPw50XW//  
AG71+eh5XYyRtalqEuirmpVpVjngYXNjerG0nltbXQEYkF3n96PLP2fy1/1wf5f18/YBU00XV/  
B8t7cSw174zL8YaA2L25Ex0uG10sX9tdS3GV+1ecIZIkg8ilCbmcTPs2H8efa1kpmjBRvskUkEttbhtsM6hpiSkI0TbtgjzUPnEkrGkFa/  
XL9lrTNKtPFxwSawXsbq/8AEWryNqrXO+KJZ9JmsljAaxhRH9tL7v0XftKkV+8v0eAxVWhWoqDjZTVrxv/  
AFsfa5LDC4jCV1JSu6btadv+fz+meJv+eL5/35/+2UV1X/CvZf8AodrP/vtf/kiivrfr+I7x/wDAf6/p+h+ffUMP2n/4F/  
X9P0P5QPirRb3Ws+I0W5aXzvE20X0UHYfTibua0zRRubIKo7bgQB02nApvWm+Iufw1+NPw+  
+J3sh+Hxld+HNT1B5fCciXs3i6HUIIbc6LEz2d+9k7bA5vYr09M2Z2jbgY8H4ovDe01yJPMG6e2DmMxNisv1NIJmTc2zeLjaFLP8AcJ3H0Az4U+  
OpPhT8XPhj8YdEh8V3Pw58rF25H4WvLbKy16P915tj3bWmopbAmKmp02nXuzLDyDuzXz+SYXBfv6LSKmlWDbUqjLfv3X8L7bn1MsXLCYxTg1  
NwtFTX3d0/8/8AM9j8aX8mqfEPV/ibZeGhofim38Ww+Jd081je3tqLa01mvdR8HPLIFWNYbdrpl1IabHgRY0L/0PL/  
bP4X+LLLxP4A8E+PrK4sCnjzSk8QtHYHliTq5FubEXDSv9sazuYbmEzrDaFvvNbpAH4V+PviprvxX+I3jf4la9p9lpN9468XxviG30Kz2vD4at7  
qCa3tdOtpkhtVv1sre4u42uBaeeJ/PU+RDsw335+wB46t9V8Eal8JtaFzG3w1g1HU/  
BbTWxto59IvLxprTzTJ3sFvMsUsNwtJ5huHRYUAZYUywrve05whyw0jyxUX9nqteP9Hluc1XGKU4811Zcq9d7/  
X4bH6sabrKyxAMHfAUZYlQM5yYrBs4Hpt00ua3Y74R0sgWB0KSRvDLKI2feFctG/lv/  
AKvDZULyW0AK4CyufFCRYgq+Wq0E2MzPHG8i43chCyKrLsPy2F7bvN/ZPKuSsUwhDk+YpYKcLjgE5B2/  
XKjrl1oRpV6cYxs2t9e3+Z9jgcVVqpSm4393aNt0r/mer+GbWzgs0MKYlerI0onjm01fNb03y/  
KIIG3rvXjPHIrtUN3Ess1wQ8UsEqzSm2eUmcfut5Rk2feYrndu04jG018myeP/  
AI1XM93F4J8JaTqFrBGNXmuNQiSDiTf5BRZbKbfny5d2HG3C5+8K4vxH4n/  
ad1a0Wz1S3/4RuAsc6Dqddp14uRvW5c28Xl+SNvKEK2fNL90d+SpU9y1GpJ2g2rppZxU9NzrzXJScVUlPbtJpPzue7N8GtA1/

```

VT4j1fSDd2tsb5p5LK5aKmm4MRiN3HNv8A+eLmBV6fvQcZwswq0C1tG+zadp9s1layXSxta28Ub2izbNouH81vML+X+7IEeXJ3Nfk974Y+MF88N
sde8W21vqRDT7vEKiKL70Vz+5htwJ/0E7ZDNH5YtgSFyUNR8GeKycQ3XiNxbFrRQIFswot/
pjpJzWuwMCQx7E8sHG3zZ0ea5q2WYxP6xWqQAprmajFLS0ezf8AXyv6GFww0jT9vWq0pKn78LGCv9tFZ97/APDHuGu3Nrp0/
wBkMkId2mbe8qkx8x5AjAcHbnJw3fuTXmt7d3M7ypDGGjQHexL2DDZC7fK0c7W6Y6cjmUutPCLKYbnvdavb6SIkxefu3j0PM3uZ23ltsfJUY2nrn
Iv6ld29rGiGRmysm1BlQY0c+Y8knzBdoK7QV07ceQ0vN0oqUHUltdVnky/
EYVYoyg1zRV17t+kenU4C5vbXTcSW4e0hsoRPqmsPcMJBdpml7JrmdRswRF5qqylk3eYh3HZivx2j+LUmi/ET4neJtI0/
Sdbh8W+JvGY06710xwW0LS9WXTTrfT7i1VvM3G0FrcsOU8zzvVvYvLy/
6IfTKffI3+GPw58QDTYBrtS8faPq2gQS6hFvLItGW1S7nsDn0ReXGVZ8vdiH940a/
HDRfMS3sLct5wgt4Yw4Xy1PzS0S1wX2E7yD8zZ2jk1y0KX1mtCuk3BSUm1stv8AgfPTU+WzbM4zw9eFWUXWLTahb3dfdtP57pbbbo+jvAiZi9mHk
M0rCSWSQL5cZkd97GOHL+Wh3YCB2AwOew/
Tn9mHWZL3xt4T0TU2t105U1K7DS0JZ0mtYYUha1TKFSPtDGQq+SFTqTX5veALViYZScdk0cY74PGc8Zx1I/Dmv0r/AGR9PL/F74X3hNqhuo/
FntLb3iCRYiINMKicAvGJi3lsuwqmzg7m5WvfwtdCXFFa61I9T4jnLP3ZNNP7L/8ij9H82P/Qbl/wDAGf8A+S6K7T+2bn/oG6R/34X/
ABOr6j2MP733/wDA/q/pbzfyw/vff/wP6v6W/kh1qK6eWRBoATiGAYFV2Iz/
ALwsRHh5M5+b5z05xxnjQ0kbbk+V4Dwxwc+Z144xt2jJyDntXq+rwnIzncb3B0WxguuBtXGcDaBnqQc9BxmF78hL03GGHf73PX80nevjcvcqTw06
d0K3yXtaWun9JHVnWhp1HKtLm54K6tJpXtdDdt7h5YwWgZHHlXcOGUxgSKecbQxz1zn7wxX3F+wjPJH8f/EHgeS4mLs/
GXw0mktWuZizPq+juZLe1s/3WYBdi+mNw4dsC3i0yTAX8IaXdAKGMW791LFt3d5tgDfdxLPLzjGW3csMc/Xf7JWu3+h/t/
s8X1ibZV1DxTL0+rLcxIQ3GmzaVeCWNXJXy/mKsSVf7in0Ye3hoRVZpX/f00/w27bIMpvF0pJu91u32X+fU/bW0lM0u6/
s3VXgju9sMzfZzhcQRWmgiXzAiB3BhLEhAGCMYBGa6K5uVZVjWTzIz17buhHGTtxnvn0e0c1btPDsOpwW1vFRRBkvYYLwyB3eVNV1J1llchMpc
CREij3YhEDESSTCef3UL14e1qTR9djk3Tc0UrBniTgCocKDNkLvm+Xcv3s14141MZU9r8NKwLtlJcu9t/1PtclmNej0Eb5Lq94q9LZb/
I9DsLwIF8yFDFb8RqTIN2/
r80bR42HbjIcYIC4wd2Rqs32lp3t7bcGUuyp9ofykjySxJnGeDnGAFpVwVoMttqNpPKs09I9mdjBmx8+0hdqk78HgfdaA0c89hp3g+w1NdPuJLu
SL7f9pJFvbgNKEj8nzVuMyLTLiRRGSDyr4zjFdNspQjCTpyaqJXi+br7vb1f9NH1mFxcK0L83vaWa01uunfTtY+c77XrqKKSyiu3ima4igliWVP
Nm18pWiZ25tu0yLTYHoTwTjGbbDX1xei08a5t74K73VnPG0uX0sJWUmPcJfnDDy12BV5b0R9ZX+ieHfdQTD0synltJIGu2W42TnafnjG1QuSCXX
0SQoLdaDXC3g0vzEudQhtrZU55VtEmAW0WVJhhd+B0E+zgLGtE5vmmLcuG8+piKzpyTmrbN6dP8v60t6P1zEUqUoxn7ttnr209NEeBzLqE7PtB0
WZY0PGNnU546kg4wMY5JPI8+8R3k8cM1p5ypJ0j2bnaCI7ibE1vG8hdfJhKEfx5kxUiPamVYnj1nxdrtpoS3kCRKJo5Z0EwL+SLGzGz92fMLl+Ru
XaFHXD8vxD8UdRvW8N+NtSu7m6ijm8K699leC4MDPMI4AJUfaxSWDcvLMASPNbGM15deTq0Z05/
DNWdtHa62fQ8fE1ZTTqSs5RStppuuH+bP7SvXvVfjN8SLj7Dqkp8CFD8SeF9IsGs/L3eIbcxR+KyJ/
OXz4N1rPbguPJkyhmwEkk5t4cvs0mC8hTsZcrnaE3cAFhn0B457c4ri9FtrhtJtHnneWaYyyyFTmUSSsVklXcd9w2D9ouN26f5MquwA+veFbUh
4SepwJH97Pv6+v5jg162V040sPyRvaUb0+r2js3rc/P8AMqcatdVJ0XNzJ2Uml9np8z60+H1rKbVDHD5mUDtzjaLFPHQ5Eu/
r1XZzv3fL+jh70nh+Zvin80L5oJha2y3dxJPE5RLAXMMUbK67SjV027hL4toH158zK/
Fmw80Mw6fu+0bVaEhj5AZ1WTJLLH5w8zy9vzAFQdw+7X6u/speHhDYavr95uuLKWx0qx0e4NttkkkjGp/bpNu+Qw+VvtPk3N5hc/
Mu3Ld0S1Z4nMVSq0YUo8qs9+rPKxn+z4atWp6TpwvG+qvdLVddz7C/4RbS/+g+f/AABP/wAnUU3Fp6z/APfo/wCNf0x1a12l9/9f0/S3x/
9q4vvD/wBf5/1f0t/LDqNi3lPuXB5x8v0zj5vb/GvH9ZsSfn2nAz/AHSe/b5v519Bat/q/wAH/wDQVrxrWpuy/
wDAV51+U4etP21PVfEuh9tjYqrRq0fZbadYr9Dkd0jcFISPVbju/wBwjbb759c19M/
AG3YfHT4GajudTZeJr5sjKxpmx+e+SbkJtLhLBQltpXI5r5103/j4i+kv/stfU/wP/wCRx8E/9hr/ANoXFe5Tr1YTjJSV019leS/z+/
yKymhTvsWtubv2UT9/fA8i3eh6LHLG6LaLeW8+2feL1or+eW3vEmCL5Wf0cGLZKMAHZ0cDoPFukW/
iSyW21LypLm13f2fd+WFlhjZ5k4bM5by4drLkC7DlWDHHLFDH/kTPDP/Xvd/wDpQa7HWP8AWQ/V/wD0BK8TF1Z060uW37347rvy7dj6b6tS/
vf+BP8Ar/
hz5x1bSvEwh6tZahps9yx01blIrGCUw294JRBK3AAkVvK8oeWcJ48x+Rk1rwwFua0W2g8Q2L3YXKL00Uum037jb5fm+eohUSbz5ew7kC4k4bcSvb
Xv/H2f+B/0rxLxj/YEr3/rLj/WuZVJLr/Wn+X9aW78E/Y1acYX5eZaN3vp/
wAA607+PPhKOK8xDq2o3Xy08szsGQpv2LFML43Z075hja0TXjetfFKLXniTT7q6hijuLnuIJYmi f92f3KRTbs/NmXc3l/JheDnA8zf/AI+bv/ro/
wD0qA/4+h9R/Om6s2mnaz8j2q9eao1GrXUDnPT/
AIP3+R3V7f3mqwuGaS1cyzRox30r0I9yvLgF921cHYoXBWduIHZD+0zrD6N8HvEetY2paQS6R9h37CE1qa3gE5uNjH90LbeIjbnzSceZGBk/
Uft/qU+g/kk+Pf2yv8AkixjP/r98P8A/pZJWdr2T2bX5nhYjGV/Y1NY/D/
Ku6Pzlt7JLS1ggVhJHbCOyWUJs80xQwzNKU3PsLfaQmzc+BHned2B6n4ThejRAL90BsdckZ+nPHUn068585b/AFQ/6/H/APSLTq9R8E/6wf8AXL/
2Y17uJisFh6EqGjmlzcc5u127Hzs37aSc+620tt/kj7N+HRsSayS48poJLRVEbKpufM3n54jb/
ALrf5uwYbedhXlXzx+137PsVrD4S0iwsrdJH1G1huQkqjBYJwrmZHUH9hbevJILEYxxX4yfb7/X6Z/19xf8AobV+0Hws+7H/ALi/
+gy1nkEuXN8KraVq8VU13um902p5+bUoxy3GSV01SbXXW6PoX+yL7/nws/8AwIT/AON0VYor9c9lt/lf/gT/AK7/ANLX8u5p/wA34Ly/4P3+R//
Z",
"additionalParams":
{
"card" : 9876,
"facility_code" : 12345,
"auto_number" : "A123AA77",
"expired" : 3,
"phone" : "0123456789",
"pin" : 33,
"is_locked" : 1
}
}

```

### 3.4 Response example:

```

{
  "Status": "SUCCESS",
  "Result": "7BC06491-C30D-450F-BF45-F9F5D680A501",
  "Description": ""
}

```

```
}
```

### 3.5 Response parameters:

Parameter	Parameter description
Status	Completed request status: SUCCESS - successful, if error - see <a href="#">List of possible errors of the Face Recognition Server</a>
Result	Added user ID in GUID format
Description	Request description

## 4 Updating user in face database

### 4.1 General request format:

POST <http://IP-address:port/firserver/UpdatePerson>

### 4.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	ID of Face Recognition Server in the face-Intellect software package
id	Yes	ID of the user in <a href="#">GUID</a> format
name	No	Name If not specified, the name will not be updated
surname	No	Surname If not specified, the surname will not be updated
patronymic	No	Patronymic If not specified, the patronymic will not be updated
department	No	Department If not specified, the department will not be updated
comment	No	Comment If not specified, the comment will not be updated
additional Params	No	Additional parameters of the <i>Intellect</i> user from the dbo.OBJ_PERSON table (see <a href="#">Base Intellect database tables</a> ) <i>Note. Text parameter values should be enclosed in double quotes.</i> Example: "additionalParams": { "visit_document": "F183825", "phone": "0123456789" } If not specified, the additional parameters will not be updated.

### 4.3 Request example:

POST <http://127.0.0.1:10000/firserver/UpdatePerson>

```
{
  "server_id" : "1",
  "id" : "FB94569E-86A6-EB11-BE5B-B06EBF811A3F",
  "name" : "John",
```

```

"surname" : "Kramer",
"patronymic" : "Kramer",
"department" : "Department 2",
"comment" : "Some comment",
"additionalParams":
{
"visit_document": "F183825",
"phone": "0123456789"
}
}

```

#### 4.4 Response **example**:

```

{
  "Status": "SUCCESS",
  "Result": "FB94569E-86A6-EB11-BE5B-B06EBF811A3F",
  "Description": ""
}

```

#### 4.5 Response parameters:

Parameter	Parameter description
Status	Completed request status: SUCCESS - successful, if error - see <a href="#">List of possible errors of the Face Recognition Server</a>
Result	Updated user ID in <a href="#">GUID</a> format
Description	Request description

## 5 Getting information about users

### 5.1 General request format:

POST http://IP-address:port/firserver/ReadPersons

### 5.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	ID of the Face Recognition Server in the <i>Face-Intellect</i> software package
id	No	Intellect user ID If not specified, the response will return information about all users
page	No	Number of results pages If not specified, then page = 1
pageSize	No	Results per page If not specified, then pageSize = 1, i.e. there will be 1 result on 1 page

### 5.3 Request example:

POST http://127.0.0.1:10000/firserver/ReadPersons

```
{
  "server_id": "1",
  "objectType": "PERSON",
  "id": [
  ],
  "page": 1,
  "pageSize": 2
}
```

### 5.4 Response example:

**Click to expand...**

```
"PersonList": [
  {
    "Id": "1CEE737C-B5D4-EB11-AC0A-B42E999B6711",
    "Name": "",
    "Surname": "1CEE737C-B5D4-EB11-AC0A-B42E999B6711",
    "Patronymic": "",
    "Department": "3",
    "Comment": "",
    "Timestamp": "2021-06-24T06:29:16.199",
    "ImageId": "1EEE737C-B5D4-EB11-AC0A-B42E999B6711",
```

```

        "AutoCreated": true,
        "LastRecognitionTimestamp": "2021-06-24T06:29:28.270",
        "RecognitionCount": 1,
    }
],
"Total": 220,
"Response": {
  "Status": "OK"
}

```

Users are listed in the response in order of time of their adding to the database (last added at first).

Using the identifier from `PersonList.Imgeld` parameter, you can get the user's photo with a separate query. See [Getting image by ID](#).

## 5.5 Response parameters:

Parameter	Parameter description
PersonList	List of users with information from the database of persons
id	User identifier from the face database
name	Name
surname	Surname
patronymic	Patronymic
department	Department
comment	Comment
timestamp	Time of adding a face to the base of reference faces in UTC+0 format
Imgeld	Reference face image identifier from the face database
AutoCreated	<b>true</b> , if the unrecognized face was added automatically
LastRecognitionTimestamp	Time of the last pass in UTC+0 format
RecognitionCount	Number of passes
Total	Total users in the face database
Response	Response
Status	Completed request status: OK - successful, if error - see <a href="#">List of possible errors of the Face Recognition Server</a>

## 6 Deleting users from the database

### 6.1 General request format:

POST http://IP-address:port/firserver/DeletePersons

### 6.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	ID of the Face Recognition Server in the <i>Face-Intellect</i> software package
id	Yes	Person ID in <a href="#">GUID</a> format * all * - to remove all faces from the reference faces database

### 6.3 Request example:

POST http://127.0.0.1:10000/firserver/DeletePersons

```
{
  "server_id": "1",
  "id": [
    "4c6ecb1d-2a37-e711-9e4b-9c5c8e763a8f"
  ]
}
```

```
{
  "server_id": "2",
  "id": [
    "*all*"
  ]
}
```

### 6.4 Response example:

```
{
  "Description": "",
  "Result": "4c6ecb1d-2a37-e711-9e4b-9c5c8e763a8f",
  "Status": "SUCCESS"
}
```

### 6.5 Response parameters:

Parameter	Parameter description
Description	Request description

Parameter	Parameter description
Result	The value of the id parameter in the request
Status	Completed request status: SUCCESS - successful, if error - see <a href="#">List of possible errors of the Face Recognition Server</a>

## 7 Searching for faces by image

### 7.1 General request format:

POST http://IP-address:port/firserver/FindFaces

### 7.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	ID of the Face Recognition Server in the <i>Face-Intellect</i> software package
findPersons	Yes	The number of matches to be returned
image	Yes	Image in the base64 format

### 7.3 Request example:

POST http://127.0.0.1:10000/firserver/FindFaces

```
{
  "server_id" : "1",
  "findPersons": 2,
  "image" : "/9j/4AAQSkZJRgABAQEAYABgAAD/
2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/
2wBDAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/
wAARCApAIEDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAEAwQFBgcICQoL/
8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdIS
UpTVFVWV1hZWMkNkZWNanaGlc3R1dnd4eXqDhIWGh4iJiPktLJWVl5iZmKqjPkwMmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+TL5
ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAEAAwQFBgcICQoL/
8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZH
SLKUI1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoq0kpaanqKmsr00tba3uLm6wsPExcBHyMnK0tPU1dbX2Nna4uPk5
ebn60nq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD85fFHjAarqevaj9u85tR1jUdTRRKTst72RPIkdCB5E03kyebbkMY9i5kbIrgLW8V5xcyX09428
yH0i25Zx947t52c4G0rNDZYPN9U1u4jsIrq4ijiv9Uub6+vvLXyBJdyvD5zpH8wjhxTMUAJE0Ww7b+0FxxG1vGH5ZnSRfLEuCM7cZ0w7upPQZwM4
5rnq4mrVkpTauyt2/yPpcvpUoYSSlf6xy/u9Xbm9zp1W+np6nun/Cdh5Jry4maIq0caRxl3eVv3EuAgTJYHBRiOetaFt45bxBdW0kKXitriQ/
azvMJYCMp5e35F+9ufPzc4r5fg1a4LLxdBvLbIqepPG0AYwAM840cYgK9L8L57Z4pZJki8qWBN23DfVfMyw+YZC+WMr79RxxWFWrVd0Shbma00vrf
+v0Pdy7B1as6cmm3ddLJ7d07d/L8bfW0s63eNZw+hWd5cFZY4GWkyg4a4UqgnwFLExhCgMbgH3+YY5+rP2a/g/
DqE2ka3NZCSxt7e+umupZAZyP4/JZYwSAVDMTv679oBAXy58IdHvvGGpuLndI7tCpa2s7W5mWwGMyfa5Fy6bMK005QWB404Y5/
Vz4WeC30i1g0nTr14bAW7reu0HL7ZpQcWwPznlXydwfyfyr08BS0+y561/rEVzQs2lZafZW/XyPtsL70hScZr964+7/
iVlt5a7372vt3ltp4jia4BuGSN/
JKHyoljkwFCxASBCoQb3BPmYX5V2jMEUuqPHDDGWG92LFj8u4rghbdv5888d+nb3tjBaW6yWYhkhiiWBybjktBnMgUqcb95+TPy7Rljzjk476Fpt1
s372M/PGV2hs9MSZ5/i/g4/KuupJ+zktNu3p/l/Wlt4VZuSttZvsLdS3ds0a7fLX5sg/xgbR2AxjJI5I59gtizBGL+0bgGGQVxnJxwM5HTp0/
Lvraj9kQiSzL553houP3R+XHzH72/
knhcBc8544+4u0jniCOZf0EhUIpIUR7Dg4JwG3jHAzgyBiub02pvNtRbW6Rl+I53a2e3EaqbkhhKV8zyxD6LtXdv8AMH05SNv8RqrxjXJYi+WK00
QNdrDc+ViIKrflFu3pv5bIXYc/Lzj06vo/7NDfackn2fzpxWcpCSAy+WEx8x6l9wH3Btx3J48Y1gWqyWzKPsly961hJ5hU/
ZVuIpX+0DBXz8G2w0f7r7wPmcYPn4+hehWq00/
brjzQvqua8fs6X0W1zlddpXm1yrWXTT1Pzn+0HhK31bRrmlu7ZI760sNdvoHLYSyyrbCw89S5WmgfNGR97YeMnkmvwx8ao+mXYhhVRDJF58dwJD+
8aQkSRGMKQBbtUBzIFM837sYXn9vvi9a3d9GbyRm41HU9M8RQaXLG+wSW0z2cd7MyhXwIXDbFV3YfzG+Zdmw/D74mqLprmpaeki/
8AEuRLNohJvMc8Ty+bIemwynGEI0NhyzcZgk16tKFa3tJPS2ivZPZersuqPAX0HW5qkP4aTLJ76WS30tdrt/
kqXhjXJeadqFtJ57yW0sU0VyquYyImMfy5G8/v05AGAnQ54/Rv8AZ7+K0sx+M/B+oJqV3aXDXNrdpdpqhEwnxby/
8AaDG5g+0tbjyxbIdnmGWf5o8Hd+UGkyzpcgnLx9GTpuzwCGwSu3B7c8ZIXmvr74QXFzb6voEyyCSysbk3bWmHSR7hY3mtkwDJAYWIKV/
3L7tw0AEIb3sFhKWGxUK9HmjU50/i0v6WPj/AHsbTxmGqax5XFK0j+y9/wCvPQ/ob/4aBvv+gprn/ggT/wCW9Ffmh/w0H4s/6E03/wDBXN/
8q6K+n+t1/wCb8PT/ACR4P9g0NPdqPtv+75ef9aw+K/EovXd/qTzSyqbeEuba3XChVaRo2JbP8ZgVh8ox0yQMnLZNSaSYn0EzWu4kDsee0c/
r7Vl6ncPFcS/
eeSMG1dmJjVxDLNItwoIfAm+0lSgyE8sESPnC4cF35k5EHMKgE5BLnaBg8fKc47c8e5r5eVFWfInzaW18z3sPhqn1i1kfwqSbSVltHt6nrGjyfaG
YgqzywxZ+7b8Y4/2WJ9z7+8eB/Dv8AaWpWRuImLQuPJjic7ZCwILNngfw/
Lt45yQMcGvmfw5CdWH+gtJcBpYkDoCqghnB5Bb80eME9TX6GfCvT7XwtpunX2r2bSl1Vo5FkDsnkjL7wEGzPmIVySWKNPQVFKLWpzhu0OaMwm1bf
4f8/16o/QMt5KUVVsvcSeuy0j6fL/
A1c+yPgVP8ADnR4Ijp3264Wae9tLiKKOKW00baZ7doWd9xYeWok8zBwfkHQ+3XH7RtLokD6jeeFNdSFj5c1vG0cSoBxNceaI3LhdwbYYLA6B+Sa
+R9J+Lun/bb9YrZtVigtciNo/L+wABwCTLIQPP558vqQZwtiuI8U+K1hQ3Vvewy99GGeK4me4jt/
tPWLvYBIVXGC29c7eRkCutSU5x5KM4zbXK9bJ+69VbVaq/zMMJjKWLxdR1pX9nL3VD3F9ndX1/03ofd2l/
FjSPEkZ3uka7JJp1wFkgR3RZIX+Zp4ZYVmkI2kqA5ZS+D8nGD1FL4we5ggaKcGRjIqMhwV2lmsSfv7s5xgbcdTmvzc8JXx0u6ht9P0+IxXCST3Mm
meapYnDR77fdNkku+zEibfmGGdcfanwoKa2unwvDdJdPhHz9tJEXeVbUxGffnZ503zYyPkk3bm/u89EYYiTSqRXI/i0tp7vX5/
```

wBXR7UJ4ZyXI5c3S7vrp/n+P3eh674otLVYhMJPtk4nPmKSQxjEf3vl0zvbv4yWzk/  
3a+cvFP7QV74UnuIrOG05LYPEN16FMJG0KQv2S5Y87idoIzsaA9MEp/G1bjwzqIvrSWOK07hmUTSIN0Uw8sLCiFskszkF8j0B8pGK/  
MrxcFrPcTeIJDxbD6kxjW5cosyrynsRht/mYYHGw7cbs1x1lea0IUp0e03azV3sn/maVakKV0dSpFyhFXkLo2r20fc+o/C3xa8baxqYvb/  
xNdvFL5j20nWduYRCrn9+sk63LebnEAQ+S3a3yndgegJ4501JJIJNWTUpJZZLhrhPhLXCyLEyOpPmWMLgzuerZwASOp+ENL+MXgjRoVsrp9YtNcs  
1ZDDa2rSQgT8RyJOGQz58r0AibABYQ3FS6+Icd6kcn0N2L60d5Jb37e8sdxbrkP5eyHcCruPM3IG+by1Hymtvqllep7jSSlo21ovh1f9fmrFKY7PMH  
G9J0qnJKydpNSs7a3tp/Xy+q/GF5EwtruxRpBYaRcWdLNI6bjGXL30wCtLmYy24KE/  
L5XLvuG38Qv94Zi0P4j6iBSca0216P7cL+SQlbaI7p447cgeX9nNwu5xM3m+aDsTan36zeG2fWNNt7vSoZTp98rxR3BjLR5Nm0XKPbyPJ5KfMn  
lv5h8wM/A2c/G37b3hb7HffDbWLe2Wyp31vSGtkImKyPDosupJ8RcuYiBAYGRuY+d8vPiSyicMfh6iqxbhUT5U/iaS6J637Ld/  
iyFHRrOpUqNGXsZx5ZSkmy2WvNb5eVuh8C6Qgkcr7dqAkSTE52FD8uExzuyx++u3HF0R9pfs7aaJ/  
Fwj6mHnaZJHeSSnGJPMtbs4aLZFljKXsbcaV2nbnduyPjPw/KI7mHzE3pcTQrJDnaRHNNcxGTftb/  
AFfk527fn8wDK7Sw+9v2f9J1JPHfw+i0m1mvtW1DwbnS9F0eOM+TftepDAzTXPziFU8xGQNBLv8AMG4da+jwmGqyxNFVivkc1zW0dt0v9fij5GLU  
lQxwLV02jktvfbLpV/8AhLNO/wcfc9/8B2/wor9Tf+GGvjf/0Lun/wDgvj/+SKK+o+oYftP/AMC/r+n6GX9o4nvD/  
wAAj5f5f1pb+YxbH42kutQldSi0440akjDZJKLxJKz7VyXGwbdnybPvNni1tpL7cXa3U3mlwtLCMqdC0hChEBz1bnnbkYzgzl7lofmXcwZmVwSyAp  
kS4+9L967NvGThs59ufcft4Pt737Abq184SzkRRyLSJDKmJntP+DjBYDbs0A3b57B4enUhgPjPm30dlftP8z1qNecqsIvls3ba39bf1pb7P/  
AGZ/gzqur+E/DVwbaK031GSe4utSnGIIiWQRDOxYGMx5bY/zebubKIUBb7rvPgXdajZy6Po13dGeKEs1xb25e0nL8v2Z/  
tC+Zt2uHwFCTl6545X4CDTNF8J6do83mrlmSXTbcnNttm2+bdCw8tli2ReUFMQjFJ8v0K/Q3wbp9zFDBeSLHZM0YwIRgSW8gIwxit8J9mKglu/  
eSmXcvk+X82VevPDlocig0mnacVJP4d1p/SR9dhYueGnFbyikvmo/  
kfj7rHwE1PRjq0+ueI9e0m4CS2drBaiek09FwxWea4kjlk4svLiKRMmlvtDgSjg7vSphB8JfC/himl8QazP4/  
intoIxY7rizktJLfwY5Z7iU3uDuUbfEJ8wxN8yBRX60Xngvw5d7bvWUa+kAn2RRWsxaxybYS0xfz8bCq7Plydz9Mccc3w+svMhTTNDsPsrPMbhgs  
NtIqkp507ZE3nFSJSOV2HOPvVccwrSai4UffS6pJPps76PT8WYYTKYUa/OLUXtH795N6aPTTT90U+Y/  
ht4T0Xwvq0V7DgkkrzLyrhB0oikwYYWmWps4LAMAPN3fctZ9BaHPqVn4k+1aYtvBclHcRq6WyhGjvQnmKkYfEwWqMQzF9/  
IUQqejsvh5otldTyhJBM4V/LMRcIYtxwjecndByDwvQcHIPEtvs08EsXi0rxKcZ+VSm5+G/2hx9ec4rb285e6+WzsvhXl/l/  
Wlvdo4KjGpBrnup7Z/rp/  
WhwHx2itdf0G0imjY7ewe4hWQ4E5MJDN5FH7wxsdUPl2feb0R8veIvCwj+MobWDWdFhvitoy2uB5ckTRLGAyzKjY3fJn5SSEHXFFYPjnS2uLS7g  
jUSiRYpLm9J27CocoiwZbcTlfg3q4440ePMNHsII5gGECgiYxyggF92QWzgLcMDDZz1G0fHqz9njqE3ooyTu/  
RL+u53V8PGVKcYpttaLfqunU+M7D4Igf6LbDq+FYPtVrLlItc3WrK0ciEqAstsd0y+wIo+aXjfyxv67p37PNjqN5bX3izw3YNGjLLHBpcwgWZlws  
blliYbYjs2qU040/3cc/  
U2n+HtPjAdoLDS28Fz5m052D5Q0ozzk8HARqHhMKRLHKmza4KrfTxb38xsjrjheh45ruxGN9pQqQpySnKNo23v7u1tfl5LUfPVMkoVpXrUfd0vZ  
uNkuXZ/1+J4Te+Gde0yykTtre0UWlq++FlV47kRBFt4oVcXCb0BkEkg80yb00wbRu/  
Mr9uSxiTwJpOrW8kd9Lp2qpfmKN+UMiJFJAGw2z0Qxk2n0wAxjg9V/FrWZSSKR4zM0oLUGHzd6xD97ECZF2hVMTNdDacRY4/  
Jz9s83F5p3hrS90wYJtZ1dNUT44NgWG5BSpSrt+8bclXyrjAMKH84gGPHzFHQnjKWZyao5txjVT1V09Laq+/m/  
kdzoUcHg61KjTioUfrTjyV2lpLbP7j8xrTSrya+halC7ZNPijuk48qVxd3IlwfvbCpThy5zu3DG2v3F/  
43k+DNN1TxxpXitXPaNks+l+Cn1jQLWkCwraf0bXic9F15dyZGRoInCeRETKrvAGT+Ndspgv7mNymZL7vqCMK+0Mb0Ly00NreFapTR03nyxBj95  
5mV/cd/gmXPawEmpPPLP8AaoFAH2e0McPnEvEdVluMgPliI8+FczBcRu44Ffp0VzeIqUfaW96SvZW6J6H5hmsVhKeKxFS04uT5vej8nT5n6z/  
wDC2fGv/P8A6j/4Mf8A7nor52/ti99NQ/8AAMf/AB6ivrPq1LTL7/6/p+lvhf7VxFeH/gC/z/q/  
pb+YfXvDtxNdeWoMqWd1cRnZsErQeV+4L7j5ZkEnD/Pt2n5DK4+0vgT8MLPwt08IQ+Frd9Q1y/  
027v760ZGitzJ4xblrNLvMqLJakuN5TMxmHyR7Pm+ZPGXmWt7dW9XI0a4b97WwMDk4BEKbyfke5LZBVthQD58/L9a/  
skftc+FA9i3wr+IHidTNE1C/  
wBc1G40TU5rCS+1gWwSWX7PY20ZjZ44P9Hw5ka9iDCeBtgCEH4XL69BYfGSacvq8HKmk9XZxXzuv6Z+sYeLgvbU9JfEvtvtHy82FzfgHRJ9GmsDq  
UE1peRXU8E2nT25UQmBoxmG5L4kvt2M+RGRtXAYH5fuzwvc+1rEZDHPF8rFvMEJic8Kp2537wTym00tfJWt+MvBNudC1C/  
wDFWmXmt+Itk1naR3LmZplLeeZorIm1FyZLaaQRu0iOpdjhc5jJb1zWjr8waXzjtEbLFCpmQuvD7mwm4yG2fKSM8HKdjgrYnDV3f2NRVPsycnyp6b  
r5H22BhhY0nKnz0V4k4tybV1y9PK3y/E+s9KmkUy7m2t5oi2F2o8QkM/  
l7+Ebevlbc88Pu3KeAvII4CIPPVobTwnUwpJw+DGMn5R+AweDjka58y0xTYfZbhLj95H5xJHR/  
M8vGefl2bSP4s7vbl9/4sjELuXc3bsxhccgZxvJABxjKfXnpznKn8cfU7I+9JRdrPTt2/  
yX9W0k8ReIdk0Q0R5GwW5x5Swb9jKXwMmTD7j9EHpx25LQobi61CJ1c3YMc0qWca5um8zacW8QJ8/btG8/  
IV+Tj5uPPp7051W6893lvwnbzi1htef3XLqW8z5c71AJ54PAwK4PxF8TvG3hDVYOPDXgHxDqL380L3NjmJ7BU2hZPtRtpPs4m30duxvME0dy7Mnu  
W69V+Z0KlCLTV7rZPetWsp5ll9a9iubec1ac3VvLGYDXdkJt82M90yBVG8pMyGqs2fubBu+evE922k3FtqmnQXKaPdx3CQSS7kllh3e3aDzfmT8EwKhN  
Ty2L0ZgzcU5ta9+0f4iX0Sof+Eo0j7XcrHLJHZPLDFzykR7xf3KdvCFhbHy9im2G/  
wa2T7nr5fX4hfFDx14rN5NbrJOWrn5LWIF4rGK1PCkNkxDdIs/zuFteYwSvy8erJepYimns2k+nSjvTV6kE9nJH2voerG/  
0+1uVkegF0V3Dcn3eD3yc8ZAxjvXSzX8cVuzSMFYidoLZyMHdzjOACD0Puk8A0aa7srcIrtFggMvJAZcg4GQBzngZ6fnoajrcxiUPChCrJk5+7kL  
2y0Dj88Z46VTowh0MLe6atdtrotV8v60tnXk1WhR05JGS01tZPfpSuxn+MdUIjvbtLgw/ZoLmYN5fnAioTd/  
FHg4PX24HGa+CPijPZatG91qMzLWMt000rXarbWEE0UM1wMnLGMogj05vNy3+r2Yb6I8deLyLS9tLSV3d4ZEm2R+dvDzW8IjKjG3cJH0cnOza  
Aeo/Kr46/EgXPxF8+fidQuLzLw14d05NMu9KiikVZ9QvIVkkdyZbTzDILHyhv3D3b5wJL8AcFV3xlnWVnU6Ljy/meHmLT2Mjpw50XW//  
AG71+eh5XYyRTalqEUirmPvPvjngYXNjerG0nlbtXQEYkF3n96PLP2fy1/1wf5f18/Ybu00XV/  
B8t7cSw174z18YaA2L25Ex0uG10sX9tdS3GV+1ecIZIkg8i1CbmcTPs2H8efa1kpmjBRvskUkEttbhtsM6hpiSkI0TbtgZjUPnEkRgkFa/  
XL9lrTNKtPFxwksawXsbq/8AEWryNqrXO+KJZ9JmsLjaAxxhRH9tL7v0XftKbV+8v0eAxVWhWoqDjZTVrxv/  
AFsfa5LDC4jCV1JSu6btadv+fz+meJv+eL5/35/+2UV1X/CvZf8AodrP/vtf/kiivrfr+I7x/wDAf6/p+h+ffUMP2n/4F/  
X9P0P5QPirrb3Ws+I0W5aXzvE20x0UhXYFtIbua0zRRubIKo7bgQB02nApvWm+Iufw1+NPw+  
+J3sh+Hxld+HNT1B5fCcis3i6HUIIbc6LEz2d+9k7bA5vYr09M2Z2jbgY8H4ovDe01yJPMG6e2DmMxNIsv1NIJmT2zeLjaFLP8AcJ3H0AZ4u+  
OpPhT8XPhj8YdEh8V3Pw58Rf25H4WvlbKy16P915tjz3bWmopAmKMP02nXuzLDyDuzXz+SYXBfv6LSK6lWDbUqjLfv3X8L7bn1MsXLCyXtG1  
NwftX3d0/8/8AM9j8aX8mqfEPV/ibZeGhofim38Ww+Jd0816je3tqLa01mvdRh8HPLIFWNYbdrpl1IabHgRY0L/0PL/  
bP4X+LLLxP4A8E+PrK4sCnjzSk8QtHYhliTq5FubEXDSv9sazuYbmEzrDaFvvNbpUAH4V+Pvivrxx+I3jf4la9p9lpN9468XXviG30Kz2vD4at7  
qCa3td0tpkhtVv1srE4u42uBaaeJ/PU+RDsw335+wB46t9V8Eal8JtaFzG3w1g1HU/  
BbTwxto59IvLxprpTzTJJsFvMsUsNwtJ5huHRYUAZyUywrve05whyw0jyxUX9nqtpe9Hluc1XGKU4811Zcqd9L/  
X4bH6sabrKyxAMHfAUZYlQM5yVyrBs4Hpt00ua3Y74R0sgWB0KSRvDLKI2feFctG/lv/  
AKvDZULYWH0AK4CyFuFCRYgW+Wq0E2MzPHG8i43chCyKrLsPy2F7bvn/ZPKuSsUwhDk+YpYKcljgE5B2/  
XKjrXl1oRpV6cYXs2t9e3+Z9jgcVvqpSm4393aNT0r/mer+GbWzgs0MKyLerI0onjm01fNb03y/  
KIIG3rvXjPHIrtUN3EsslwQ8UsEqzSm2eUMcfut5Rk2feYrndu04jG018mYEP/  
AI1XM93F4J8JaTqFrbGNXmuNqisDiTf5BRZbKbfny5d2HG3C5+8K4vxH4n/  
ad1a0Wz1S3/4RuAsc6DqdpM14uRvW5c28Xl+SNvKEK2fNl90d+SpU9yk1GpJ2g2rpPzXU9NzrzXJScVULpBtJpPzue7N8GtA1/  
VT4j1f5Dd2tsb5p1L5aKmm4MRiN3HNv8A+eLmBV6fvQcZswq0C1tG+zadp9s1layXsxta28Ub2izbNouH81vML+X+7I2EeXJ3NFk974Y+MF88N  
sde8W21vqRDT7vEKiKL70Vz+5htwJ/0E7ZDNH5YTGvSFyUNR8GeKyCQ3XinxBFRQIfsWot/

pjPjzWuwMCQx7E8sHG3zZ0ea5q2WYxP6xWqQaprmaJFLS0ezf8AXyv6GFwW0jT9vWq0pKn78LGCv9tFZ97/APDHuGu3Nrp0/  
wBkMkId2mbe8qxk8x5AjaChbnJw3fuTXmt7d3M7ypDGGjQHexl2DDZC7fk0c7W6Y6cjmUutPCLKYbnVdabv6SIkxefu3j0PM3uZ23ltsfJUY2nrn  
Iv6ld29rGiGRmysm1BlQY0C+Y8knzBdoK7QV07ceQ0vN0oqUHUltDVnky/  
EVYVoyg1zRV17t+kenU4C5vbXTtcSW4e0hsoRPqmsPcMjBDpml7JrmdRswRF5qqylk3eYh3HZivx2j+LUmi/ET4neJtI0/  
Sdbh8W+JvGY06710xwW0LS9WXTTrfT7i1VvM3G0FrcsOU8zzVwYvLy/  
6IfftKfFi3+GPw58QDITYbRtS8faPq2gQS6hFvliTGW1S7nsZDn0RexGZV8vdiH940a/  
HDRfMS3sLct5wgt4Yw4Xy1PzS0SIwX2E7yD8zZ2jk1y0KX1mtCuk3BSUm1stv8AgfPTU+WzbM4zw9eFWUXWLTahb3dfdf57pddbo+jvAizI9mHk  
M0rCSWSQL5cZkd97GOHL+Wh3YCB2AwOew/  
Tn9mHWZL3xt4T0TU2t105U1K7DSoJZ0mtYYUha1TKFSPTDGQq+SFTqTX5veALViYZSCdk0cY74PGc8Zx1I/Dmv0r/AGR9PL/F74X3hNqhuo/  
FNtLb3iCRYiINMkicAvGJi3lsuwqmgz7m5WvfwtdCXFFa61I9T4jnlP3ZNtNPN7L/8ij9H82P/Qbl/wDAGf8A+S6K7T+2bn/oG6R/34X/  
ABor6j2MP733/wDA/q/pbzFYw/vff/wP6v6W/kh1qK6eWRBoATiGAYFV2Iz/  
ALwsRHn5M5+b5z05xxnjQ0kbbk+V4Dwxc+Z144xt2jjJyDntXq+rwNIZncb3B0WxguuBtXGcDaBnqQc9BXMf78hl03GGHf73PX80nevjcvqTw06  
d0K3yXtaWun9JHvNWhp1HKtLm54K6tJpXtDddTdt7h5YwWGZHhLXcOGUxgSkecbQxz1zn7wxX3F+wjPJH8f/EHgeS4mls/  
GXw0mktWuziZPq+juZLe1s/3WYBdi+mNw4dsC3i0yTAX8IaXdAKGMW791LFT3d5tgDfdxLPLzjGW3csMc/Xf7JWu3+h/tT/  
s8X1ibZV1DxTL0+rLcxIQ3GmzaVeCWNXJXy/mKsSVf7in0Ye3hoRVZpX/f00/w27bIMpvF0pJu91u32X+fu/bw0lml0u6/  
s3VXgj9sMzfZzhcQRWmgjXzAiB3BhLEhAGCMYBGA6K5uVZVjWtZIZ17buhHGTtxnvnoe0c1btPDSOpwW1vFbRRBkvYYLwyB3eVNV1J1llchMpc  
CREij3YhEDESCTCef3U1L4e1qTR9djk3Tc0UrBnImTgcocKdnKlvm+Xcv3s14141MZU9r8NKwltLJcu9t/1PtcLmNej0Ebx5Lq94q9LzB/  
I9DsLwIF8yDFb8RqTIN2/  
r80bR42hBjIcYIC4wd2Rqs32lp3t7bcGUuyp9ofykyjSxJnGeDnGAF1PvWvoMttqNpPKs09I9mdjBmx8+0hdqk78HgfDAa0c89hp3g+w1NdPuJLu  
SL7f9pJFvbgnKEj8znVuMyLTLIRRGSDyr4zjFdnSpQJCTpyaqJXi+br7vb1f9NH1mFxcK0L83vaWa01uunfTtY+c77XrqKKCSyiu3ima4igliWVP  
NmI8pWiIZ5tu0yLtYHoTwTjGbbDXei08a5t74K73VnPG0ux0sJWUmPcJfnDDy12BV5b0R9ZX+ieHfDqDTd0synltJIGu2W42TnafNjG1QuSCXX  
OSQoLdaDXC3g0vzEudQhtrZU55VtEmAW0WVJhhd+B0E+ZgLGtE5vmnlCuG8+piKzpyTm2rbN6dP8v60t6P1zEUqUoxn7ttnr209NEeBzLqE7PtB0  
WZY0PGNnU546kg4wMY5JPI8+8R3K8cM1p5ypJ0j2bnaCI7ibElvG8hdfJhKEfx5kUipamVYnjlnxdrtpoS3KCRKJo5Z0EwL+SLGzGz92fMLL+Ru  
XaFHXD8vxD8UdRvW8N+NtSu7m6ijm8K699leC4MDPMIA4JUFaxSWDcvLMASPNbGM15deTq0Z05/  
DNWdtHa62fQ8fE1ZTTqSs5RStppuuH+pB7SvXvVfjN8SLj7Dqkp8CfD8SeF9IsGs/L3eIbcxR+KyJ/  
OXz4N1rpBguPJkyhmWEKkt5t4cvs0mC8HtZscraE3cAFhn0B457c4ri9FtrhtJtHnneWaYyyyyFTmUSSsVklXcd9w2D9ouN26f5MquwA+veFbU9  
4SepwpJH97Pv6+v5jg162V040sPyRvaUb0+r2js3RC/P8AMqcatdVJOXNzJ2Uml9np8z60+H1rKbVDHD5mUDtzjaLFPHQ5Eu/  
r1XZzv3fL+jH70nh+Zvin80L5oJha2y3dxJPE5RLAXMMUBK67SJV027h14tohI58zK/  
Fnw80Mw6fu+0bVaEhj5AZ1WTJLLH5w8zy9vzAFQdw+7X6u/speHhDYavr95uuLKWx0qx0e4NttkkkjGp/bpNu+Qw+VvtPk3N5hc/  
Mu3Ld0S1Z4nMVSqW5YVUo8qs9+rPKxn+z4atWp6TpwvG+qvdLVddz7C/4RbS/+g+f/AABP/wAnUU3Fp6z/APfo/wCNf0X1a12L9/9f0/S3x/  
9q4vvd/wBf5/1f0t/LDqNi3lPuX85x8v0zj5vb/GvH9ZsSfN2nAz/AHSe/b5v519Bat/q/wAH/wDQVrxrWPuy/  
wDAv51+U4etP21PVfEuh9tjYqrRq0fZbadYr9Dkd0jCfISPvBju/wBwj759c19M/  
AG3YfHT4GajudTZeJr5sjKxpmx+E+SbkJtLhLbQLtpXI5r5103/j4i+kv/stfU/wP/wCRx8E/9hr/ANoXFe5Tr1YTjJSV019leS/z+/  
yKymTvSWtubv2UT9/fa8i3eh6LHLG6LaLeW8+2feL1or+eW3vEmCL5Wf0cGLZKMAH50cDoPFukW/  
iSyW21LypLm13f2fd+wFlhjzF5kb4bM5by4dr1kC7D1WDHHLFDH/kTPDP/Xvd/wDpQa7HWP8AWQ/V/wD0BK8TF1Z060uW37347rvy7dj6b6tS/  
vf+BP8Ar/  
hz5x1bSvEwh6tZahps9yx01blIrGCuW294JRBK3AAkVvK8oeWcj48x+Rk1rwwFua0W2g8Q2l3YXKL00Uum037jb5fm+eohUSbz5ew7kC4k4bcSvb  
Xv/H2f+B/0rxLxJ/yEr3/rLJ/WuZVJLr/Wn+X9aW78E/Y1acYX5eZaN3vp/  
wAA607+PPHOK8xDq2o3Xyo8szsGQpv2LFML43Z075hja0TXjetfFKLXniTT7q6hijuluIJYmi92f3KRTbs/NmXc3l/JheDnA8zf/AI+bv/ro/  
wD0Qa/4+h9R/Om6s2mnaz8j2q9eao1GrXUdNPT/  
AIP3+R3V7f3mqquwGaS1cyrRox30rOI9yvLgF921cHYoXBWduIHZd+0zrD6N8HvEetY2paQS6R9h37CE1qa3gE5uNjH90LbeIjbnzSceZGBk/  
UFT/qu+g/kk+Pf2yv8AkiXjP/r98P8A/pZJWdr2T2bX5nhYjGV/Y1NY/D/  
Ku6Pzlt7JLS1ggVhJHbCOyWUJJs80xQwzNKU3PsLfaQmzc+BHned2B6n4ThejRAL90BsdckZ+nPHUn068585b/AFQ/6/H/APSLTq9R8E/6wf8AXL/  
2Y17uJisFh6EqGjmlzcc5ui27Hzs37aSc+620tt/kj7N+HRsSayS48poJLRVEbKpufM3n54jb/  
ALrf5uwYbedhXlxzx+137PsVrd4S0iwsrdJH1G1huQkqjBYJwrmZHUh9hbvejILEYxxX4yfb7/X6Z/19xf8AobV+0HwS+7H/ALi/  
+gy1nkEuXN8KraVq8VU13um902p5+bUoxy3GSV01SbXXW6PoX+yL7/nws/8AwIT/AON0VYor9c9lT/lf/gT/AK7/ANLX8u5p/wA34Ly/4P3+R//  
Z"

## 7.4 Response example:

```
{
  "FaceList": [
    {
      "quality": 0.756927,
      "eyeLeftX": 30.8689,
      "eyeLeftY": 48.5863,
      "eyeRightX": 57.8069,
      "eyeRightY": 48.1852,
      "faceRectLeft": 12,
      "faceRectTop": 32,
      "faceRectWidth": 64,
      "faceRectHeight": 64,
      "ProcessedPersons": 70,
      "ProcessedPersonsDuration": 0,
      "PersonList": [
```

```

{
  "Id": "6C1069EB-1037-E711-9E4B-9C5C8E763A8F",
  "Name": "Hathaway",
  "Surname": "Anne",
  "Patronymic": "",
  "Department": "0",
  "Comment": "",
  "Timestamp": "2017-05-12T12:45:49.279",
  "ImageId": "6E1069EB-1037-E711-9E4B-9C5C8E763A8F",
  "AutoCreated": false,
  "LastRecognitionTimestamp": "2021-06-25T11:17:33.909",
  "RecognitionCount": 177,
  "Sim": "0.54350495338439941"
}
]
}
]
}

```

## 7.5 Response parameters:

Parameter	Parameter description
FaceList	List of faces
quality	Face image quality (from 0 to 1)
eyeLeftX	Left eye horizontal coordinates
eyeLeftY	Left eye vertical coordinates
eyeRightX	Right eye horizontal coordinates
eyeRightY	Right eye vertical coordinates
faceRectLeft	Face coordinates relative to the left edge of the frame
faceRectTop	Face coordinates relative to the top edge of the frame
faceRectWidth	Face width in pixels
faceRectHeight	Face height in pixels
Processed Persons	Number of generated vectors
Processed PersonsDuration	Time spent generating vectors
PersonList	A list of users with information from the faces database (see <a href="#">Getting information about users</a> ).
id	User identifier from the faces database
Name	Name

Parameter	Parameter description
Surname	Last name
Patronymic	Patronymic
Department	Department
Comment	Comment
Timestamp	Time of adding a face to the reference face database in UTC+0 format
ImageId	Identifier of the reference face image from the faces database
AutoCreated	<b>true</b> if the unrecognized face was added automatically
LastRecognitionTimestamp	Time of the last pass in UTC+0 format
RecognitionCount	Number of passes
Sim	The degree of similarity of the face in the image in the request and the reference face from the face database

## 8 Getting image by ID

### 8.1 General request format:

GET http://IP-address:port/firserver/GetImage/{server\_id}/{id}

### 8.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	ID of the Face Recognition Server in the <i>Face-Intellect</i> software package
id	Yes	the image ID

### 8.3 Request example:

GET http://127.0.0.1:10000/firserver/GetImage/1/7D02839C-CCB6-EA11-8E71-74D435D0E19B

### 8.4 Response example:

JPEG image

## 9 Getting analytic data

### 9.1 General request format:

POST <http://IP-address:port/firserver/GetAnalytic>

### 9.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	Identifier of the Face Recognition Server in <i>Face Intellect</i>
minAge	No	Minimum age of persons
maxAge	No	Maximum age of persons
dateTimeFrom	Yes	Time in UTC+0 format, starting from which the data should be obtained
dateTimeTo	Yes	Time in UTC+0 format, up to which the data should be obtained
genders	No	Gender of persons whose data should be obtained: <b>0</b> - Unknown, <b>1</b> - Males, <b>2</b> - Females
faceMode	No	Type of persons whose data should be obtained: <b>0</b> - all faces, <b>1</b> - only recognized faces, <b>2</b> - only unrecognized faces
threshold	No	Threshold value of the faces similarity degree
minPasses	No	Filter by the minimum number of passes
maxPasses	No	Filter by the maximum number of passes
cameraIds	No	Camera IDs in <i>Face Intellect</i>

### 9.3 Request example:

POST <http://127.0.0.1:10000/firserver/GetAnalytic>

```
{
  "server_id": "1",
  "minAge": 0,
  "maxAge": 1000,
  "dateTimeFrom": "2018-08-17T21:00:00.000",
  "dateTimeTo": "2018-09-18T20:59:59.000",
  "genders": [
    0,
    1,
    2
  ],
  "faceMode": 0,
  "threshold": 0.8,
  "minPasses": 1,
  "maxPasses": 10000,
  "cameraIds": [
    "1"
  ]
}
```

```
]
}
```

## 9.4 Response example:

```
{
  "PassAndLikesList": [
    {
      "MainPass": {
        "Id": "53952CAF-3BBB-E811-8D01-005056C00008",
        "Timestamp": "2018-09-18T12:09:32.848",
        "Camera": "1",
        "Gender": 1,
        "Age": 31.4502
      },
      "BestPerson": {
        "Id": "5CC71799-48BB-E811-8D01-005056C00008",
        "Name": "John",
        "Surname": "Wick",
        "Patronymic": "",
        "Department": "Department 1",
        "Comment": "",
        "Timestamp": "2021-06-25T10:39:55.974",
        "ImageId": "98B73FA9-A1D5-EB11-AC0A-B42E999B6711",
        "AutoCreated": false,
        "LastRecognitionTimestamp": "2021-06-25T11:59:52.185",
        "RecognitionCount": 375,
        "revisions": [
          {
            "source": "8A91CD97-62D3-EB11-AC0A-B42E999B6711",
            "revision": "97B73FA9-A1D5-EB11-AC0A-B42E999B6711"
          }
        ]
      },
      "Sim": 0.850095
    },
    {
      "LikePasses": [
        {
          "Id": "1E1483A5-41BB-E811-8D01-005056C00008",
          "Timestamp": "2018-09-18T12:52:08.987",
          "Gender": 1,
          "Age": 31.4372,
          "Camera": "1"
        },
        {
          "Id": "F10CD78D-41BB-E811-8D01-005056C00008",
          "Timestamp": "2018-09-18T12:51:33.907",
          "Gender": 1,
          "Age": 31.4372,
          "Temperature": 0,
          "Emotion": 5,
          "Glasses": 2,
          "FacialHair": 6,
          "HairColor": 1,
          "HairType": 3,
          "Headwear": 13,

```

```

    "Liveness": 1,
    "Evasion": 1
  },
  ],
  "Total": 40,
  "Males": 40,
  "Females": 0
  },
  ],
  "ProcessedTotal": 1346,
  "ProcessedDuration": 83,
  "CompareCount": 18555,
  "Response": {
    "Status": "SUCCESS"
  }
}

```

## 9.5 Response parameters:

Parameter	Parameter description
PassAndLikesList	List of passes of unique faces
MainPass	Data on the first pass of a unique face (for more details, see <a href="#">Getting the recognition protocol</a> )
BestPerson	Data about the face from the face database and the degree of similarity of the reference face with a unique face, if such a person exists (for more details, see <a href="#">Getting information about users</a> )
Timestamp	Time of adding a face to the reference face database in UTC+0 format
ImageId	Identifier of the reference face image from the faces database
AutoCreated	<b>true</b> if the unrecognized face was added automatically. <b>false</b> if it was added manually
LastRecognitionTimestamp	Time of the last pass in UTC+0 format
RecognitionCount	Number of passes
LikePasses	List with data of similar passes (similar to MainPass). Cannot be larger than 5 elements
Temperature	Facial temperature in degrees Celsius
Emotion	Emotion: 0 - undefined, 1 - rage, 2 - fear, 3 - happiness, 4 - without emotion, 5 - sadness, 6 - astonishment
Glasses	Glasses: 0 - undefined, 1 - tinted glasses, 2 - without glasses, 3 - glasses
FacialHair	Facial hair: 0 - undefined; 1 - beard, 2 - bristle, 3 - imperial beard and mustache, 4 - imperial beard, 5 - mustache, 6 - absent
HairColor	Hair color: 0 - undefined, 1 - brunet, 2 - blonde, 3 - chestnut hair, 4 - ginger, 5 - gray
HairType	Type of bald head: 0 - undefined, 1 - bald, 2 - with receding hairline, 3 - without bald head

Parameter	Parameter description
Headwear	Headwear: 0 - undefined, 1 - cap, 2 - bandana, 3 - beret, 4 - beanie, 5 - trapper hat, 6 - fur hood, 7 - glasses on head, 8 - hat, 9 - helmet, 10 - hood, 11 - kepi, 12 - headscarf, 13 - without headwear, 14 - peaky cap
Liveness	Artificial face: 0 - undefined, 1 - no, 2 - yes
Evasion	Face concealment: 0 - undefined, 1- face not covered, 2 - mask is on, 3 - tinted glasses, 4 - face partially covered, 5 - balaclava, 6- other mask, 7 - mask is not on the right place
Total	Total number of unique faces
Males	Number of males
Females	Number of females
Processed Total	Total number of processed faces
Processed Duration	Execution time of the faces processing in milliseconds
CompareCount	Total number of the face comparisons
Response	Response
Status	Completed request status: SUCCESS - successful, if error - see <a href="#">List of possible errors of the Face Recognition Server</a>

## 10 Getting analytic data by time

### 10.1 General request format:

POST <http://IP-address:port/firserver/GetAnalytcByTime>

### 10.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	Identifier of the Face Recognition Server in <i>Face Intellect</i>
minAge	No	Minimum age of persons If not specified, then age is not taken into account
maxAge	No	Maximum age of persons If not specified, then age is not taken into account
dateTimeFrom	Yes	Time in UTC+0 format, starting from which the data should be obtained
dateTimeTo	Yes	Time in UTC+0 format, up to which the data should be obtained
cameraIds	No	Camera IDs in <i>Face Intellect</i> If not specified, then results for all cameras are displayed
genders	Yes	Gender of persons whose data should be obtained: <b>0</b> - Unknown, <b>1</b> - Males, <b>2</b> - Females If parameter value is empty, then all results are displayed regardless of gender
returnLocalTime	No	Time format of the face capture: <b>0</b> - time in UTC+0 format, <b>1</b> - local time of the Server If not specified, then returnLocalTime=0
timePeriod	No	Type of grouping the analytic data: <b>0</b> - by hour, <b>1</b> - by day, <b>2</b> - by month, <b>3</b> - by year If not specified, then timePeriod=0

### 10.3 Request example:

POST <http://127.0.0.1:10000/firserver/GetAnalytcByTime>

```
{
  "server_id": "1",
  "minAge": 20,
  "maxAge": 30,
  "dateTimeFrom": "2019-02-25T21:00:00.000",
  "dateTimeTo": "2019-03-29T20:59:59.000",
  "cameraIds": [
    "1"
  ],
  "genders": [
```

```

    0,
    1,
    2
  ],
  "returnLocalTime" : 0,
  "timePeriod" : 0
}

```

## 10.4 Response **example:**

[Click to expand...](#)

```

{
  "Rows": [
    {
      "timestamp": "2019-02-27T180000.000",
      "value": [
        {
          "age": 27,
          "gender": {
            "males": 10,
            "females": 9,
            "unknown": 0
          },
          "smile": {
            "males": 3,
            "females": 1,
            "unknown": 0
          },
          "angry": {
            "males": 1,
            "females": 2,
            "unknown": 0
          },
          "fear": {
            "males": 1,
            "females": 2,
            "unknown": 0
          },
          "happiness": {
            "males": 1,
            "females": 1,
            "unknown": 0
          },
          "neutral": {
            "males": 2,
            "females": 1,
            "unknown": 0
          },
          "sadness": {
            "males": 1,
            "females": 1,
            "unknown": 0
          },
          "surprise": {

```

```

    "males": 1,
    "females": 1,
    "unknown": 0
  }
}
]
}
]
"Response": {
  "Status": "SUCCESS"
}
}

```

## 10.5 Response parameters:

Parameter	Parameter description
Rows	List of unique faces
timestamp	First pass time in UTC+0 format
value	List of additional face characteristics (for more details, see <a href="#">Getting the recognition protocol</a> )
Response	Response
Status	Completed request status: SUCCESS - successful, if error - see <a href="#">List of possible errors of the Face Recognition Server</a>

## 11 List of possible errors of the Face Recognition Server

### 11.1 General request format:

POST http://IP-address:port/firserver/FaceErrorDescriptions

### 11.2 Request example:

POST http://127.0.0.1:10000/firserver/FaceErrorDescriptions

### 11.3 Response:

```
{
  "FAILED_TO_ADD_PERSON": "Failed to add person. Check request parameters or database",
  "FACE_NOT_FOUND": "Face not found on supplied image",
  "MAXIMUM_FIRS": "Maximum FIR in database",
  "SERVER_NOT_READY": "Server is initializing, please wait",
  "FAILED_TO_DELETE_PERSON": "Failed to delete person. Check request parameters or database",
  "FAILED_TO_ADD_PERSON_MANY_FACES": "The image shown has more than one face",
  "FAILED_TO_SAVE_DATABASE": "Failed to save person. Check request parameters or database",
  "FAILED_TO_UPDATE_PERSON": "Failed to update person. Check request parameters or database"
}
```

### 11.4 Response parameters:

Parameter	Parameter description
FAILED_TO_ADD_PERSON	Failed to add person. Check request parameters or database
FACE_NOT_FOUND	Face not found on supplied image
MAXIMUM_FIRS	Maximum FIR in database
SERVER_NOT_READY	Server is initializing, please wait
FAILED_TO_DELETE_PERSON	Failed to delete person. Check request parameters or database
FAILED_TO_ADD_PERSON_MANY_FACES	The image shown has more than one face
FAILED_TO_SAVE_DATABASE	Failed to save person. Check request parameters or database
FAILED_TO_UPDATE_PERSON	Failed to update person. Check request parameters or database