



Face PSIM. REST API

Face PSIM 1.0.1 (english)

Last update 12/05/2022

Table of Contents

1	General information.....	5
2	Getting the recognition protocol	6
2.1	General request format:.....	6
2.2	Request parameters:.....	6
2.3	Request example:.....	7
2.4	Request headlines:.....	7
2.5	Response example:	8
2.6	Response parameters:	9
3	Adding/updating user in face database.....	12
3.1	General request format:.....	12
3.2	Request parameters:.....	12
3.3	Request example:.....	13
3.4	Request headlines:.....	16
3.5	Response example:	17
3.6	Response parameters:	17
4	Updating user in face database.....	18
4.1	General request format:.....	18
4.2	Request parameters:.....	18
4.3	Request example:.....	19
4.4	Response headlines:	19
4.5	Response example:	20
4.6	Response parameters:	20
5	Getting information about users.....	21
5.1	General request format:.....	21
5.2	Request parameters:.....	21
5.3	Request example:.....	21
5.4	Request headlines:.....	21
5.5	Response example:	22
5.6	Response parameters:	23

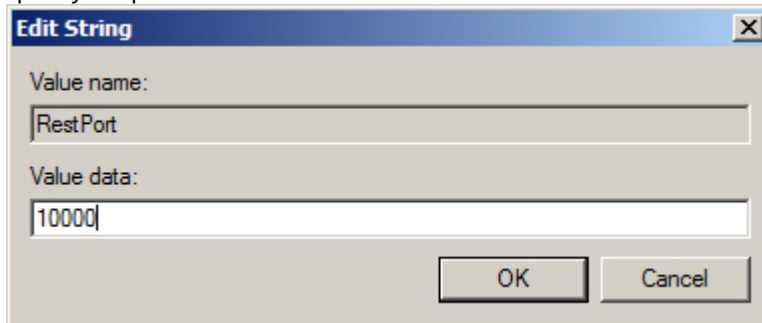
6	Deleting users from the database	24
6.1	General request format:.....	24
6.2	Request parameters:.....	24
6.3	Request example:.....	24
6.4	Request headlines:.....	24
6.5	Response example:	25
6.6	Response parameters:	25
7	Searching for faces by image.....	26
7.1	General request format:.....	26
7.2	Request parameters:.....	26
7.3	Request example:.....	26
7.4	Request headlines:.....	29
7.5	Response example:	29
7.6	Response parameters:	30
8	Getting image by ID.....	32
8.1	General request format:.....	32
8.2	Request parameters:.....	32
8.3	Request example:.....	32
8.4	Request headlines:.....	32
8.5	Response example:	32
9	Getting analytic data.....	33
9.1	General request format:.....	33
9.2	Request parameters:.....	33
9.3	Request example:.....	33
9.4	Request headlines:.....	34
9.5	Response example:	34
9.6	Response parameters:	36
10	Getting analytic data by time	39
10.1	General request format:.....	39
10.2	Request parameters:.....	39

10.3 Request example:.....	39
10.4 Request headlines:.....	40
10.5 Response example:	40
10.6 Response parameters:	42
11 List of possible errors of the Face recognition server	43
11.1 General request format:.....	43
11.2 Request example:.....	43
11.3 Response:	43
11.4 Request headlines:.....	43
11.5 Response parameters:	44

1 General information

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

1. Select the HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\AxxonSoft\PSIM\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 <i>Face PSIM</i>
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
cameras	No	Camera IDs (GUID or ID in the <i>Axxon PSIM</i> objects 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

Parameter	Required	Parameter description
dateTimeFrom	Yes	The initial boundary of the time period in UTC+0 format: yyyy-mm-ddThh:nn:ss.fff
dateTimeTo	Yes	Ending boundary of the time period in UTC+0 format: yyyy-mm-ddThh:nn:ss.fff

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 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request
Content-Type	Returns the type of received request

Headline	Headline description
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

2.5 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+yqerPT6Ghjw/
Vhy9PpMQvWHqmb2U5VQ8kic/vWIS/
b0xXNq8kJ7hPU7MGT2qdxw9FdAJPFxwLxGLUQ9wBJCPLPiB71UMoU8vhbHP0Thqb3VaSg92qH0PH0j1zyEda
a9uL0xvVfWq71oy0o8twbWu3g+mD0SQvw8C/
KhvEjElrspZFf9Q7xsu6FLolLmLa492n3tvCyBl7wNLL8+n+6RvRQnCbzDDhw9EnCAPdSsmT2PsHa8n+6HvTd
6Cj0qT3U9oFQMveIDNj0m7sU8eih0vYGvgT3INic9tWfXvevQnr1QMmG92WGHvf7ULzxxz74a8aCmlvXAa6j0v
EM08ApXTvGsovLw0uzC9TpGt00eUfbzB0F4+SUGqPQpEvbyr+K08NJQTve1zZLrt6kQ9lvwEvZCE/
j0R9B693z+APJK7l7x6K+W9yPTm08ai1jp630U90KUUPujXET0WnPa8DIW7PE32Z73Fkke8kYNmvZg/
Lb2bv+o95BUKvYRH47yr70S8ojApvaMX5bwC4Co9AuwNvSFoWtYxa09ueWFvPZ+RT3w7Mk7HnsLvDsmBD32P
AW9IuphvSZLpTyBN5W9+e4sPcuAXr2RNI288I5hvYam3LteHnm9FFVfVio8VL0ishw+QHjDPGGbhr2sJZK8E3
a5PYsDK73kgKu8cN4/
PB3Z8z10AE+9HWm7PRJhdr5l7Tm9LjRHvZtTA7wTWT+08nGa4PXmX0bs1mwk8xrGcPBgNfzyfF7A6vYPPaRxU
L3jk8e7Y3K0vX5i77z543o9Xh/9PF0eAD24hrG8VYl1vRQnhrxGpcm8uu0C0dmLAz0z2AG+0cChu6l/
Yz1Xdla9fHPnvAZuiz2dv9g7hLbYu47LJb1fUFM9IpIcvsRgBr7RC4Y9GgFCPeuYTLyw2oe9wUZNPZAPGr0G1
EK96x2nvY29ybyQdZE9/
v4jvcPhXbu9D309YwkwPYfTTD4S3fc9BtZqu3cfMLydfbu9Igsj01PYMjx0rUA9Z1qgvSmWSj3BBlm8DcuvPW
BNHr0e0YA9r5eIPMbslL26mIi852wHPvVfOD1wWPK8gHECvSVsgr3bM7i9GNleu+Lbfj0UDY29VU0hvv3xUzx
```

```

aiAY9DtccPfk+BD0M+iy9IZABvRr35Dz0psy9G7VWPVmkoD26GAE9JBptvbtnL73QTFA9+fs3veiXx7v76o09
m130040zRT3zog09YTimvAln071mTac8BPmCPcVQ8LzJ/
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"
}

```

2.6 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

Parameter	Parameter description
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
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
faceVector	Face vector data
facePattern	Face pattern (vector)
usedSDK	The used SDK: 1 – Cognitec, 2 – VISIONLABS, 3 – TEVIAN, 4 – STC
usedAlgorithm	The used face recognition and search algorithm: 0 – ALG1, 1 – ALG2, 2 – PRECISE
hits	The recognized face information from the faceDB and <i>Axxon PSIM</i> 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>Axxon PSIM</i>
sim	Similarity rate of the captured face with the image from the reference face database
surname	Last name

Parameter	Parameter description
name	First name
patronymic	Patronymic name
comment	Comment
department	Department
Status	Completed request status: OK — successful, if error — see List of possible errors of the Face recognition server
person_guid	Identifier of the corresponding <i>Axxon PSIM</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 <i>Face PSIM</i>
object Type	No	Object type
id	No	ID of the user in GUID 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

Parameter	Required	Parameter description
createPersonInAxxonPSIM	No	The flag for creating a user in <i>Axxon PSIM</i> . If the parameter is equal to 1, the user will be created in <i>Axxon PSIM</i> on behalf of the <i>Axxon PSIM</i> operator If not specified, the <i>Axxon PSIM</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)
additionalParams	No	If createPersonInAxxonPSIM = 1, you can specify the <i>Axxon PSIM</i> user parameters from <code>dbo.OBJ_PERSON</code> table (see Base Axxon PSIM database tables) <i>Note. Parameter text values must be enclosed in double quotes</i> For example: <pre>"additionalParams": { "auto_number": "A123AA77", "pin": 33, }</pre>

3.3 Request example:

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

```
{
  "server_id" : "1",
  "objectType" : "PERSON",
  "id" : "BC06491-C30D-450F-BF45-F9F5D680A501",
  "name" : "John",
  "surname" : "Kramer",
  "patronymic" : "",
  "department" : "Department 1",
  "comment" : "Some comment",
  "createPersonInAxxonPSIM": 1,
  "image": "/9j/4AAQSkZJRgABAQEAYABgAAD/
2wBDAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/
2wBDAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQEBAQH/
wAARCApAIEDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/
8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3O
Dk6Q0RFRkdISUpTVFVWV1hZmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx
8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/
8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2N
```

zG50kNERUZHSElKU1RVVldYwVpjZGVmZ2hpanN0dXZ3eHl6go0EhYaHiImKkp0ULZaXmJmaoq0kpaanqKmqsr00tba3uLm6wsPEX
cbHyMnK0tPU1dbX2Nna4uPk5ebn60nq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD85fFHjAarqevaj9u85tR1jUdTRRKTst72RPIkd
CB5E03kyebbkMY9i5kbIrgLW8V5xcyX09428yH0i25Zx947t52c4G0rNdZyPN9U1u4jsIrrq4ijiv9Uub6+vvLXyBjdyvD5zPH8wj
hxtMUAJE0Ww7b+OfXxG1vGH5ZnSRfLEuCM7cZ0w7upPQZWM45rnrq4mrVkpTauuyt2/yPpcvpUoYSSlF6xy/
u9Xbm9zp1W+np6nun/Cdh5Jry4maIqocArqxL3eVv3EuAgTJYHBRi0etaFt45bxBdWOKKXitriQ/
aZvMJYCMp5e35F+9ufPzc4r5fg1a4LxdBvLbiQepPG0AYwAM840cYgK9L8L57Z4pZJki8qWBN23DfvfMyy+YZC+WmR79RxFWFrV
d0Shbma00vrf+v0Pdy7B1as6cmm3ddLJ7d07d/L8bFw0s63eNZw+hwd5cFZY4GWKyG4a4UqgneWfLEeXhCgMbgh3+YY5+rP2a/g/
DqE2ka3NZCSxt7e+umupZAZyp4/JZYwSAVDMT6790BaxXy58IdHvVGGpULnDI7tcca2s7W5mWWMGMyfa5Fy6bMK005QWB404Y5/
Vz4WeC30i1g0nTr14bAW7reu0HL7ZpCQwPznlXydwfyf08BS0+y561/rEVzQs2LzafZW/XyPtsL70hScZr964+7/
iVlt5a7372vt3ltp4jii4BuGSN/
JKHYoLjkWfCxASBcoQb3BPmYX5V2jMEUuQPHDDGWG92LFJ8u4rgrbdv5888d+nb3tjBaw6yWYhkhiiWBybjktBnMgUqcb95+TPy7R
ljzjk476Fpt1s372M/PVG2hs9MSZ5/i/g4/KuupJ+zktNu3p/l/Wlt4VZuSttZvsLdS3ds0a7fLX5sg/
xgbR2AxjJI5I59gtizBGL+0bgGGQVxnJxwM5HTp0/Lvrajq9kQiSzL553houP3R+XHZH72/
knhcBc8544+4u0jniCOZf0EhUipIUR7Dg4JwG3jHAzG4yBiuB02pvNtrBw6Rl+I53a2e3EaqbkhhKV8zyxD6Ltxdv8AMH05SNv8Q
rxjxJYi+WK00QNdRdc+ViIKrflFu3pv5bIXYc/Lzj06vo/
7Ndfackn2fzPwCpCSAy+Wez8x6l9wH3Btx3J48Y1GwqyWzKpsly961hJ5Hu/ZVuIpX+0DBXz8G2w0f7r7wPmcYPn4+hehWq00/
brjzQvqua8f56X0W1zlddpXm1yrWXTT1Pzn+OHhK31bRrm1u7ZI760sNdvoHLySyrbcw89S5WmgfNGR97YeMNMvXw8ao+mXYhh
VRDJF58dwdJ+8aQkSRGMQBtUBZIfM837sYXn9vvi9a3d9GyBRm41HU9M8RQaXLG+wSw0z2cd7MhXwIXDbFV3YfzG+ZdmW/
D74mqLprmpaeki/
8AEuRLNohJvMc8Ty+bIemwynGEIONhyzckeZgK16tKfa3tJPS2ivZPZersuqPAX0HW5qkP4aTLJ76WS30tdrt/
kqXhjxJeadqFtJ57yW0sU0VyuYyImMfy5G8/v05AGANQ54/Rv8AZ7+K0sx+M/B+oJqV3aXDXNRPdhPplqHEWnxby/
8AaDG5g+0tbjyxIdnmGwf508Hd+UGkyzpcgnLx9GTpuzwCGwSu3B7c8ZIxmvR74QXFz6voEyyCSysbk3bWmHSR7hY3mtkwdJAY
wwikV/3L7tw0AEIb3sFhKWGxUK9HmjU50/i0v6WPj/AHsbTxmGqax5XFK0j+y9/wCvPQ/ob/4aBvv+gprn/ggT/wCw9FfmH/
w0H4s/6E03/wDBxN/8q6K+n+t1/wCb8PT/ACR4P9g0NPdqpTtv+75ef9aw+K/E0vXd/
qTZSyqbeEuba3XChVaR2JbP8ZgVh8ox0yQmnlZNSaSYn0Ezww4kDsee0c/r7Vl6ncPFcS/
eeSMG1dmJjVxDLNItwoIfAm+0LSgyE8sESPnC4cF35k5EHMKGE5BLnaBg8fKc47c8e5r5eVFWfInzaW18z3sPhqn1iLkfwqSbSVL
tHt6nrGjyfaGyqquzywxZ+7b8Y4/2WJ9zz7+8eB/Dv8AaWpWRuImLQuPJJjc7ZCwILNfgw/
L45yQMcGvmfw5CdWH+gtJcBpYkDoCqghnB5Bb80eME9TX6GfCvT7XwtpunX2r2bSl1Vo5FkDsnkjL7wEGzPmIVySWKnPQVFKLWp
zh0U0aMwm1bf4f8/16o/QMt5KUVVsvcSeuy0j6fL/
Aic+yPgvFP8ADnR4Ijp3264Wae9tLiKK0KWO0baZ7dowd9xYeWok8zBwfKqH+3XH7RtLokD6jeeFNdSFj5c1vG0cSoBxNceaI3Lh
dwbYYLA6B+Sa+R9J+Lun/bb9YrZtVigtciNo/L+wAbwTLIQPP558vgQZwTiuI8U+K1hQ3VvewyW99GGeK4me4jt/
tPWLBYIvXGC29c7eRkCutSU5x5KM4zbxK9bJ+69VbVaq/zMMJjKWLxdR1pX9nL3VD3F9ndX1/03ofd2l/
FjSPEKz3uka7JjP1wFkgR3RZIX+Zp4ZYVmkI2kqA5ZS+D8nGD1FL4we5ggaKcGRjIqMhwV2LmsSfv7s5xgbcdTmvzc8JXx0u6ht9
P0+IXCST3MmeapYnDR77fdNkku+zEibfmGGdcfanwoKa2unwvDdJdPhew9tJExeVbUxGffnZ503zYyPk33bm/
u89EYyITsQRXI/i0tp7vX5/wBR7UJ4ZyXI5c3S7vpr/n+P3eh674otLVYhMJPTk4nPMKSQxjEf3v10zvbv4yWzk/
3a+cvFP7QV74UnuIr0G05LYPEN16FMJG0KQv2SY87idoIzsA9Mep/G1bjwzPivrSW0K07hmUTSIN0Uw8sLciFskszkF8j0B8pGK/
MrxRc3FrPcTeIJDdX6kxjW5cosyrynksRht/mYYHGw7cbs1x1ea0Ip0qe03azV3sn/maVakKV0dSpFyhFXklo2r20fc+o/
C3xa8baxqYvb/
xNdvFL5j20nWduYRCrn9+sk63LebnEAQ+Sm3a3yndgeGJ4501J1JNWTUpJJLhRhpLhXCYLEyopPmWMLgzuerZwASOp+ENL+MXgj
RoVsrp9YtNcs1ZDDa2rS0gT8RyJOGQz58r0AibABYQ3FS6+Icd6kc0N2L60d5Jb37e8sdxBRkp5eyHCruPM3IG+by1Hymtvqlp
7jSSlo21ovh1f9fmrFKY7PMHG9J0qnJkydpNSs7a3tp/Xy+q/GF5EwtruxRpBYaRcWdLNI6bjGXL30wCtLtmYy24KE/
L5XLvuG38Qvj94Zi0P4j6ibSca0216P7cL+SQLbqaI7p447cgeX9nNwu5xM3m+aDsTaN36zeG2fWNNt7vSoZTP98rxR3BjLR5Nm0
XKPbyPJ5kFmnlv5h8wM/A2c/
G37b3hb7HfDbWLe2WytP31vSGtkImKyPdospJ8RcuYiBAyGRuY+d8vPiSyicMfh6iqxbhUT5U/iaS6J637Ld/
iyfHRr0pUqNGXsZ5ZSkm3y2W8N5eVuh8C6Gqkcr7dqAKSTE52FD8uExzuyx++u3HF0R9pfs7aaJ/
FWjm6HnaZJHeSSnGJPMtbS4aLZFLjKSXbcAV2nbnduyPjPw/KI7mHzE3pcTQRJDNaRHNNcxGTFtb/
AFFk527fn8wDK7SW+9v2F9J1JPHfw+i0m1mvTW1DwbnS9F0e0M+TftepDAZTPXziFU8xGQNBLv8AmG4da+jwmGqyxNFVivkc1zW0
dt0v9fij5GLUQxWLV02jktVfblPV/8AhLNO/wCfC9/8B2/wor9Tf+GGvjF/0Lun/wDgvj/+SKK+o+oYftP/AMC/
r+n6GX9o4nvD/
wAAj5f5f1pb+YbxH42kutQldSio44o4kjdZJkLxJKz7VyXGwbndybPvNnitpl7cXa3U3mlwtLcMqdC0hChEBz1bnnbkYzglz1z7of
mXcwZmVwSyApkS4+9l967NvGThs59ufcft4Pt737Abq184SzkZRRYlsJDKmJntP+tdjBYDbs0A3b57B4enUhgPjPm30dlfTp8z1
qNecqsIvls3ba39bf1pb7P/AGZ/gzqur+E/DVwbaK031GSe4utSnGIIiWQRdOXyGMx5bY/
zebukIUBb7rvPgXdajZy6Po13dGeKES1xb25e0nL8v2Z/
tC+Zt2uHwFC7L6545X4CDTNF8J6do83miRLmSXTbcnNttm2+bDcwf8tli2ReUfMqjFJ8vOK/
Q3wbp9zFDBesLHZM0YwIRgSW8gTwxit8J9mKGLu/eSmXcvK+X82VevPD1ocig0mnaCvJP4d1p/SR9dhYueGnFbyikvmo/
kfj7rHwE1PRjq0+ueI9e0m4CS2drBaieK09FwxWea4kjl4svLiKRmMlvtDgSjG7vSPH8JfC/hi1mL8QazP4/
intoIxY7rizktJLFWY5Z7iU3uDMUbFEJ8wxN8ybRX60Xngvw5d7bvWUa+kAn2RRWsaxymbYS0xfz8bCq7Plydz9Mccc3w+svMhT
NDsPrPMbhgsNtIqkp507ZE3nFSS0V2H0PvVccwrSai4UfF56pJPps76PT8WYTYKyuA/0LUXtH795N6aPTTT90U+Y/
hT4T0XwvQ07DgkkrzLrYhB00ikwYYWVmwPs4LAMAPN3fcTZz9BAHPqVn4k+1aYtvBcLHCRq6WYhgjvQnmKkyfEwWqGMQzF9/
IUqQejsvh5otldTyhjBM4V/LMRciYdtxwjecndByDvwQcHIPEtvs08EsBxI0rXcZ+VsM5+G/2hx9ec4rb285e6+WzsvhXl/L/
Wlvdo4KjGpBrnunp7z/rp/
WhwHx2itdf0G0imjYy6ewe4hWQ4E5MJdn5FH7wxsdupL2feb0R8veIvCwj+MobWDWdFhvitoy2uB5ckTRLGAyzKjY3fJn5SSEHXF
fYPjnS2uL57gJUSiRYpLm9J27CocoiwZbcTLgf3q4440ePMNHsII5gGecgiXyggf92QWzgLcMDDZz1G0fHzq9njqE3ooyTu/
RL+u53V8PGVKcYpttaLfqnuU+M7D4I6lBdQ+FYPtVrLILtc3WrK0ciEqAstsdOy+wIo+aXjfyxv67p37PNjqN5bX3izw3YNGJL
LHBpcwZLwSbllYbYjsqU040/3cc/
U2n+HtPjAdoLds28Fz5m052D5Qozzk8HArqHhMKRLHKmza4KrFtxjb38xsjrjheh45ruxGN9pQqQpySnKNo23v7u1tfl5LuFPVM
koVpXrUfd0vZuNkuXZ/1+J4Te+GdE0yykTtre00UWlq++FLV47kRBFt4oVCxCB0BkEkG80yb00wRu/

Mr9uSxiTwp0rW8kd9Lp2qpfmKN+UMiJFJAGw2z0Qxk2n0wAxjgj9V/FrWZSSKR4zM0oLUGHzd6xD97ECZF2hvMTnDdACrY4/
Jz9s83F5p3hrS90wYJtZ1dNUt44NgWG5bSPsrT+8bcLXyrjaMKH84gPHzfHQnjKWZYao5txjVT1V09Laq+/m/
kdzoUcHg61KjTiouFrTjV2lplDpb7j8xrTSrya+ha1C7ZNPijUk48qVxd3IlfvbcPThY5zu3DG2v3F/
4Jk+DNN1TxrpxiTxPaNqs+l+Cn1jQLWkCWraFq0bXiC9F15dyZGRoInCeRETkrvAGT+Ndspgv7mNmYLZV7qCMK+0Mb0Ly00NrEfa
PtR03nyxBj955mV/cD/
gmXPawEmpPLP8AaofAH2e0McPnEvEdVlUmguPLI8+FcZbcRu44Ffp0VzeIqUfaW96SvZW6J6H5hmsVhKeKxFS04uT5vejD8nT5
n6z/wDC2fGv/P8A6j/4Mf8A7nor52/ti99NQ/8AAMf/AB6ivrPq1LTL7/p+lvhf7VxfeH/gC/z/q/
pb+YfXvDTXndeWoMqWd1cRnZsErQeV+4L7j5ZkEnD/Pt2n5Dk4+0vgT8MLPWT08IQ+FrD9Q1y/
027v760ZGitiZJ4xblrNLvMqLJakuN5TMxmHyR7Pm+ZPGXmWt7dW9xIIo4b97WMDk4BEKbyfkE5LZBVthQD58/L9a/
sKftC+FPA9i3wr+IHIDTNE1C/
wBc1G40TU5rCS+1gWwSWX7PY20ZjZ44P9HW5ka9iDceBtgCEH4XL69BYfGSacvq8HKmk9XZxZxuv6Z+sYelgvBU9JfEvtvtHy82f
ZfgHRJ9GmsDQUE1peRXU8E2nT25UQmBoxmG5L4kvt2M+RGRtXAYH5fuzwvc+c1rEzDHPF8rFvMEJiC8Kp2537wTym00tfJWt+Mv
BNudC1C/
wDFWmXmt+Itk1naR3lmZpLeeZorIm1FyZLaaQRu0iOpdjhc5jJb1zjwr8waXzjtEbLFCpmQuvD7mwm4gY2fKSM8HkDjgrYnDV3f2
NRVPSycnyp6br5H22BhhY0nKnz0V4k4tybV1y9PK3y/E+s9KmkuY7m2t5oi2F2o8QkM/
l7+Ebевlbc88Pu3KeAvII4CIPPVobtnWuWpJw+DGMn5R+AweDjka58y0vXTYfZbhlj95H5xJHR/
M8vGefl2bSP4s7vb19/4sjELuXc3bsxhccgZxvJABxjKfXnpznKn8cfU7I+9JRdrPTt2/
yX9WOk8ReIdK0QR5GWW5x5Swb9jKxWmMtd7j9EHpx25LQobi61CJ1c3YmC0qWca5um8zacw8QJ8/btG8/
IV+Tj5uPp7051W6893lvWnbzI1htef3XLqW8z5c71AJ54PAwK4PxF8TvG3hDVYOpDXgHxDqL380L3NjmJ7BU2hZPtRtpPs4m30d
uxvME0dy7MnuW69V+Z0KLC1LTV7rzPetWsp5lLa9iubeC1ac3VvLgyDXdkJt82MH90yBG8pMyGQsFubBu+evE922k3FtqmnQXKaPd
SXCQSS7k1ke3adZfMt8EwKhntY2L0JgzckU5ta9+0f4i0Xsof+e0oj7XcrHLJHZPLFDfzykr7xf3kdvCFhbHy9im2G/
wa2T7nIr5fX4hffDx14rN5NbrJowrn5LWIF4rGK1PCKnkxDdIs/zuFteYwSvy8eRjEpYimns2k+nSjvTV6kE9nJH2voerG/
0+1uVKEgFOV3Dcn3eD3yc8ZAxjvXsZx8cVuzSMFYIdoLZyMHdzjOACD0PuK8A0aa7srcIrtFggMvJAZcg4GQBzngZ6fnoajrcxiU
PcHCrJk5+7kL2y0Dj88Z46VTowh0Mle6atdtrotV8v60tnXk1Whr05J6S01tZPfp8uxn+MdUIjvbtLgw/ZoLmYN5fnAiOtd/
FHg4PX24HGa+CPijPZatG91qMsLWMtv000rXarBwCE0oUM1wMmnlGMogj05vNy3+r2Yb6I8deLyL9S9tL5V3d4ZEm2R+dvDzW8IjK
jG3cJH0cn0zaAeo/Kr46/EgXPx8FQ+FIdQuLzwl4d05NMu9KiikvZ9QvIVkkumhtZDIHLhyvj3Db5wJL68AcFV3xLNVvN6Lyj/
meHmLT2MjPw50XW//AG71+eh5XYyRTaLqEUirmPvPvjngYXNjerG0nlbtXQEyKf3n96PLP2fy1/1wf5f18/YBu00XV/
B8t7cSW174z18YaA2l25Ex0uG10sX9tdS3GV+1ecIZIkg8i1CbmcTPs2H8efA1kpmjBRvskUkEttbhtsM6hpiskIOTbtgtzUPnEk
RgkFa/XL9lrTNktPFxwKsawXsbq/8AEwryNqrX0+KJZ9JmsLjaAxxhRH9tL7v0XftKbv+8v0eAxVwhwoqDjZTVrvxv/
AFsfa5LDC4jCV1J5u6btadv+fz+meJv+eL5/35/+2UV1X/CvZf8AodrP/vtf/kiivrfr+I7x/wDAf6/p+h+ffUMP2n/4F/
X9P0P5QPirrb3ws+I0W5aXzvE2oX0UHXyFtIbua0zRRubIKo7bgQB02nApvwm+Iufw1+NPw+
+Js3h+Hxld+HNT1B5fCccixs3i6HUIIbc6LEz2d+9k7bA5vYr09M2z2g2bjgy8H4ovDe01yJPMG6e2DmMxNIsv1NIJmTc2zeLjaFLP
8AcJ3H0Az4u+OpPhT8XPhj8XYdeH8V3Pw58Rf25H4WvLlbKy16P915tj3c3bWmopbAmKmp02nXuzLDyDuzXz+SYXBfV6LSkM6LWDb
UqzlfV3X8L7bn1MsXLcytG1NwtfTX3d0/8/8AM9j8aX8mqfEPV/
ibZehofim38Ww+Jd08I6je3tqlA01mvdRh8HPLIFWNYbdrpl1IabHgRY0L/OPL/
bP4X+LLLxP4A8E+PrK4scnjzSk8QtHYhilitQ5FubEXDSv9sazuYbmEzrDaFvvNbpAH4V+Pvivrpx+I3j4la9p9lpN9468Xxvi
G30Kz2vD4at7qCa3td0tpkhtVv1sre4u42uBaaeJ/PU+RDsw335+wB46t9V8EaL8JtaFzG3w1g1HU/
BbTWxt059IvLxprTPZTJJsFvMsUsNwtJ5huHRYUAZYUYwrvE05whyw0jyxUX9nqtep9Hluc1XGKU4811ZcqD9l/
X4bH6sabrKyxAMHfAUZYlQM5yVyrBs4Hpt00ua3Y74R0csyGB0KSRvDLKI2feFCTg/lv/
AKvDZULyWHOAK4CyufuFCRyqW+Wq0E2MzPHG8i43chCyKrLsPy2F7bvn/ZPKuSsUwhDk+YpYKcLjgE5B2/
XKjrXlloRpV6cYXs2t9e3+Z9jgcVVqpSm4393aNt0r/mer+Gbwzgs0MKyLerI0onjm01fnb03y/
KIIG3rvXjPHirtUN3Ess1wQ8UsEqzSm2eUMcfut5Rk2feYrndu04jG018myeP/
AI1XM93F4J8JaTqFrbGNXmuNqisDiTf5BRZbKbfny5d2HG3C5+8K4vxH4n/
ad1a0Wz1S3/4RuAscX6DqdpM14uRvW5c28Xl+SNvEK2fnL90d+SpU9yk1GpJ2g2rppZxU9NzrzXJScvUlpBtJpPzue7N8Gta1/
VT4j1fSDd2tcsb5p5LK5akMm4MRiN3HNv8A+eLmBV6FvQCZwSqw0C1tG+zadp9s1layX5xta28Ub2izbNouH81vML+X+7I2EeXJ3N
fk974Y+MF88Nsde8W21vqRDT7vEKiKL70Vz+5htwJ/OE7ZDNH5YtGSFYUNR8GeKYcQ3XinXbFrRQIfsWot/
pjpPzWuwMCQx7E8sHG3zZ0ea5q2WYxP6xWqQaprmajFLS0ezf8AXyv6GFwW0jT9vWq0pKn78LGCv9tFZ97/APDHuGu3Nrp0/
wBkMkId2mbe8qxk8x5AjAcHbnJw3fuTxmT7d3M7ypDGGjQhexL2DDZC7fK0c7W6Y6cjmUtPCLKYbnVdab6Sikxefu3j0PM3uZ2
3ltsfJUY2nrnIv6ld29rGigRmysm1BLQY0C+Y8knzBdoK7QV07ceQ0vN0oQUHUltdVnky/
EVYVoyg1zRv17t+kenU4C5vbXTtcSW4e0hsoRPqmsPcMJBdpml7JrmdRswRF5qqylk3eYh3HZivx2j+LUmi/ET4neJtI0/
Sdbh8W+JvGY06710xwW0LS9WXTfrT7i1VvM3G0Frcs0U8zVvYvLy/
6iftKffI3+GPw58QDTybrtS8faPq2gQ56hFvLitGW1S7nsZDn0RexGZV8vdiH940a/
HDRfMS3sLct5wgt4Yw4Xy1PzSOSIwX2E7yD8z2jkiy0KX1mtCuk3BSUm1stv8AgfPTU+WzbM4zW9eFWUXWLTahb3dfdt57pddb
+jvAizI9mHkM0rCSWSQL5cZkd97GOHL+Wh3YCB2Aw0ep/
Tn9mHWZL3xt4T0TU2t105U1K7DSoJZ0mtYYUha1TKFSPtDGQq+SFTqTX5veALViyZSCdk0cY74PGc8Zx1I/Dmv0r/AGR9PL/
F74X3hNqhuo/FntLb3iCRYiINMkicAvGJi3lsuwmqzgm75WvfwCDXFFa61I9T4jnlP3ZnTNNP7L/8ij9H82P/Qbl/
wDAGf8A+S6K7T+2bn/oG6R/34X/ABor6j2MP733/wDA/q/pbzfyw/vff/wP6v6w/kh1qK6eWRBoAtiGAYFV2Iz/
ALwsRHn5M5+b5z05xxnjQ0kbbk+V4Dwxw+Z144xt2jJyDntXq+rwNIzncb3B0WxguuBtXGcDaBnqQc9BxmF78hL03GGHf73PX8
0nevjcqvTw06d0k3yXtaWun9JHvNWhP1HKtLm54K6tJpXtDdt7h5YwWZGHhLXc0GUGxgSkecbQxz1zn7wxX3F+wjPJH8f/
EHgE54mls/GXw0mktWuzizpW+juzLe1s/3WYBdi+mNw4dsC3iOyTAX8IAxdAKGMW791LFt3d5tgDfdxLPLzjGW3csMc/
Xf7JWu3+h/t/s8X1ibZV1DxTLorLcxIq3GmzaVeCWNXJxy/mKsSVf7in0Ye3horVZpX/f00/w27bIimpvF0pJu91u32X+FU/
bw0lm0u6/
s3VXgju9sMzfZzhcQRwMgiGxZaIb3BhLEhAGCMYBGA6k5uVZVjWtZiz17buHGtTxnvn0e0c1btPds0pw1vFbRRBkvYYLwyB3eV
NV1Jl1lchMpcCREij3YhEDESSTCef3U114e1qTR9djks2C0UrBnimTgcocKdnKlvm+Xcv3s14141MZU9r8NKWlTLJcu9t/
1PtclMNej0Ebx5Lm94q9LzB/I9DsLWIF8yFDFB8RqTIN2/
r80bR42hBjIcYIC4wd2Rqs32lp3t7bcGUuy9pofykjySxJnGednGafLpvWvoMttqNpKs09I9mdjBmx8+0hdqk78HgfdAA0c89hp
3g+w1NdPuJLuSL7f9pjFvbgNKEj8nzVuMyLTLiIRGSDyr4zjFdNSpQjCTpyaqJXi+br7vb1f9NH1mFxcK0L83vaWa01uunfTtY+c

```

77XrqKKCSyiu3ima4igliWVPNmi8pWiIZ5tu0yLtYHoTwTjGBBdX1xei08a5t74K73VnPG0ux0sJWUmPcJfnDDy12BV5b0R9ZX+i
eHfDqDtD0synlTJIGu2W42TnafNjG1QuSCXXOSQoLdaDXC3g0vzEudQhtrZU55VtEmAW0WVJhhd+BOE+ZgLGt5vmnlCuG8+piKz
pyTm2rbN6dP8v60t6P1zEUqUoxn7ttnr209NEeBzLqE7PtBOWZY0PGNnu546kg4wMY5JPI8+8R3k8cM1p5ypJ0j2bnaCI7ibELvG
8hdfJhkEFx5kxUiPamVYnj1nxdrtpoS3kCRKJo5Z0Ewl+SLGzGz92fMLL+RuXaFHXd8vxD8UdRvW8N+NtSu7m6ijm8K699leC4MD
PMI4AJUfaxSWDcvLMASPNbGM15deTq0Z05/DNwdtHa62fQ8fE1ZTTqS5SRStppuuh+bP7SvxVvfjN8SLj7Dqkp8CFD8SeF9IsGs/
L3eIbcxR+KyJ/
OXz4N1rpBguPJkyhmwEkkt5t4csv0mC8hTsZcrnaE3cAFhn0B457c4ri9FtrhtJtHnneWaYyyyyFTmUSSsVklXcd9w2D9ouN26f5
MquwA+veFbUh4SepwpJH97Pv6+v5jg162V040sPyRvaUb0+r2js3rc/
P8AMqcatdVJ0XNzJ2Uml9np8z60+H1rKbVDHD5mUDtzjaL fPHQ5Eu/
r1XZzv3fL+jH70nh+Zvin80L5oJha2y3dxJPE5RLAXMMUbK67SJv027h14tohI58zK/
Fnw80Mw6fu+0bVaEhj5AZ1WTJLLH5w8zy9vzAFQdw+7X6u/speHhDYavr95uuLkwx0qx0e4NttkkkjGp/bpNu+Qw+VvtPk3N5hc/
Mu3Ld0S1Z4nMVSqW5YVUo8qs9+rPKxn+z4atWp6TpwvG+qvdLVddz7C/4RbS/+g+f/AABP/wAnUU3Fp6z/APfo/
wCNf0x1a1L2l9/9f0/S3x/9q4vvd/wBf5/1f0t/LDqNi3lPuXB5x8v0zj5vb/GvH9ZsSfN2nAz/AHSe/b5v519Bat/q/wAH/
wDQVrxrWpuy/wDAv51+U4etP21PVfEuh9tjYqrRq0fZbadYr9Dkd0jcfISPvBju/wBwjbb759c19M/
AG3YfHT4GajudTZeJr5sjKxpmxEe+SbkJtLhLBQLtpXI5r5103/j4i+kv/stfU/wP/wCRx8E/9hr/ANoXFe5Tr1YtJJSV019leS/
z+/yKymhTvSWtubv2UT9/fA8i3eh6LHLG6LaLeW8+2feL1or+eW3vEmCL5Wf0cGLZKMAH0cDoPFukw/
iSyW21LypLm13f2fd+WFlhjFz5kb4bM5by4drLkC7D1WDHHLFDH/kTPDP/Xvd/wDpQa7HWP8AWQ/V/
wD0BK8TF1Z060uW37347rvy7dj6b6tS/vf+BP8Ar/
hz5x1bSvEwh6tZahps9yx01blIrGCUw294JRBk3AAKvV8KoeWCj48x+Rk1rwwFua0W2g8Q2l3YXKL00Uum037jb5fm+eohUSbz5e
w7kC4k4bcSvbXv/H2f+B/0rxLxJ/yEr3/rLJ/WuZVJLr/Wn+X9aw78E/Y1acYX5eZa3vp/
wAA607+PPhK0K8xDq2o3Xyo8szsGQpv2lFML43Z075hja0TXjetfFKLXniTT7q6hijuLnuIJYmi92f3KRTbs/NmXc3L/
JheDnA8zf/AI+bv/ro/wD0QA/4+h9R/Om6s2mnaz8j2q9eao1GrXUdNPT/
AIP3+R3V7f3mqwuGaS1cyzRox30r0I9yvLgF921cHYoXBwDuIHZD+0zrD6N8HvEetY2paQS6R9h37CE1qa3gE5UnjH90LbeIjbn
zSceZGBk/UFt/qU+g/kk+Pf2yv8AKiXjP/r98P8A/pZJWdr2T2bX5nhYjGV/Y1NY/D/
Ku6Pz1t7JLS1ggVhJHbCOyWUJs80xQwzNKU3PsLfaQmzc+BHned2B6n4ThEjRAL90BsdckZ+nPHUn068585b/AFQ/6/H/
APSLtq9R8E/6wf8AXL/2Y17uJisFh6EqGjmlzcz5u127Hzs37aSc+620tt/kj7N+HRsSayS48poJLRVEbKpufM3n54jb/
ALrf5uYwbedhXlxz+137PsVrD4S0iwsrdJH1G1huQkqiBYJwrmZHUh9hbevJILEYxxX4yfb7/X6Z/19xf8AobV+0HwS+7H/ALi/
+gyInkEuXN8KraVq8VU13um902p5+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 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request
Content-Type	Returns the type of received request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

3.5 Response **example**:

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

3.6 Response parameters:

Parameter	Parameter description
Status	Completed request status: SUCCESS — successful, if error — see List of possible errors of the Face recognition server
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 <i>Face PSIM</i>
id	Yes	ID of the user in GUID 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

Parameter	Required	Parameter description
additionalParams	No	<p>Additional parameters of the <i>Axxon PSIM</i> user from the <code>dbo.OBJ_PERSON</code> table (see Base Axxon PSIM database tables)</p> <p><i>Note. Text parameter values should be enclosed in double quotes.</i></p> <p>Example:</p> <pre>"additionalParams": { "visit_document": "F183825", "phone": "0123456789" }</pre> <p>If not specified, the additional parameters will not be updated.</p>

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 headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request

Headline	Headline description
Content-Type	Returns the type of received request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

4.5 Response **example**:

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

4.6 Response parameters:

Parameter	Parameter description
Status	Completed request status: SUCCESS — successful, if error — see List of possible errors of the Face recognition server
Result	Updated user ID in GUID 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 <i>Face PSIM</i>
id	No	<i>Axxon PSIM</i> 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 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request

Headline	Headline description
Content-Length	Shows the size of characters in bytes, received in the request
Content-Type	Returns the type of received request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

5.5 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.ImageId** parameter, you can get the user's photo with a separate query. See [Getting image by ID](#).

5.6 Response parameters:

Parameter	Parameter description
Person List	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
ImageId	Reference face image identifier from the face database
AutoCreated	true , 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 List of possible errors of the Face recognition server

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 <i>Face PSIM</i>
id	Yes	Person ID in GUID 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 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request

Headline	Headline description
Content-Type	Returns the type of received request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

6.5 Response **example**:

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

6.6 Response parameters:

Parameter	Parameter description
Description	Request description
Result	The value of the id parameter in the request
Status	Completed request status: SUCCESS - successful, if error - see List of possible errors of the Face recognition server

w0H4s/6E03/wDBxN/8q6K+n+t1/wCb8PT/ACR4P9g0NPdqPtv+75ef9aw+K/E0vXd/
qTzSyqbeEuba3XChVaRo2JbP8ZgVh8ox0yQMnLZNSaSYnOEzWu4kDseeOc/r7Vl6ncPFcS/
eeSMG1dmJjVxDLNItwoIFAm+0LSgyE8sESPn4Cf35k5EHMKgE5BLnaBg8fKc47c8e5r5eVFWfInzaW18z3sPhqn1iLkfwqSbSVL
tHt6nrGjyfaGyGquzywxZ+7b8Y4/2WJ9zz7+8eB/Dv8AaWpWRuImLQuPJJjc7ZCwILNfGfw/
Lt45yQMCGvmfw5CDWH+gtJcBpYkDoCqghnB5Bb80eME9TX6GfCvT7XwtpunX2r2bSl1Vo5FkDsnkjL7wEGzPmIVySWKnPQVFKLWp
zh0U0aMwM1bf4f8/16o/QMT5KUVVsvcSeuy0j6fL/
AIC+yPgvFP8ADnr4Ijp3264Wae9tLiKKOKW00baZ7doWd9xYeWok8zBwfkhQ+3XH7RtLokD6jeeFNdSFj5c1vG0cSoBxNceaI3Lh
dwbYYLA6B+Sa+R9J+Lun/bb9YrZtVigtciNo/L+wABwTLIQPP558vgQzWtIuI8U+K1hQ3VvewyW99GGeK4me4jt/
tPWLvYIvXGC29c7eRkCutSU5x5KM4zbXK9bJ+69VbVaq/zMMJjKWLxdR1pX9nL3VD3F9ndX1/03ofd2l/
FjSPEKz3uka7JJp1wFkgR3RZIX+Zp4ZYVmkI2kqA5ZS+D8nGD1FL4we5ggaKcGRjIqMhwV2LmsSfv7s5xgbcdTmzvc8JXX0u6ht9
P0+IxXCST3MmearYnDR77fdNkku+zEibfmGGDcfanwoKa2unwvDdJdPHezW9tJExeVbUxGffnZ503zYyPk3bm/
u89EYYiTSqRXI/i0tp7vX5/wBXR7UJ4ZyXI5c3S7vvp/n+P3eh674otLVYhMJpTk4nPmKSQxjE3f3vL0zbv4yWzk/
3a+cvFP7QV74UnuIR0G05LYPEN16FMJG0KQv2S9Y87idoIzsA9Mep/G1bjwzPivrSWOK07hmUTSIN0Uw8sLCiFskszkF8j0B8pGK/
MrxRc3FrPcTeIJDxBd6xkjW5cosyrynksRht/mYYHGw7cbs1x1ea0Ip0qe03azV3sn/maVakKV0dSpFyhFXklo2r20fc+o/
C3xa8baxqYvb/
xNdvFL5j20nWduYRCrn9+sk63LebnEAQ+Sm3a3yndgegJ4501JIJNWTUpJJJLhrhplhXCYLEyopPmWMLgzuerZwASOp+ENL+MXgj
RoVsrp9YtNcs1ZDDa2rS0gT8RyJOGQz58r0AibABYQ3FS6+Icd6kc0N2L60d5Jb37e8sdxBrkp5eyHcCruPM3IG+by1Hymvtvqlep
7jSSlo21ovh1f9fmrFKY7PMHG9J0qnJkydpNsS7a3tp/Xy+q/GF5EwtruxRpBvARcWdLNI6bjGXL30wCtLtmYy24KE/
L5XLvuG38QvJ94Zi0P4j6ibScA0216P7cL+SqLbqaI7p447cgeX9nNwu5xM3m+aDsTaN36zeG2FWNNt7vSoZTp98rxR3Bjlr5Nm0
XKpbyPJ5KfMnlv5h8wM/A2c/
G37b3hb7HffDbWLe2Wytp31vSGtkImKyPdosupJ8RcuYiBAYgRuY+d8vPiSyicMfh6iqxbhUT5U/iaS6J637Ld/
iyFHRr0pUqNGXsZx5ZSkM3y2WvNb5eVuh8C6Qgkcr7dqAkSTE52FD8uExzuyx++u3HfOR9pfs7aaJ/
FWjm6HnaZJHeSSnGJPMtbs4aLZFLjK5XbcAV2nbnduyPjPw/KI7mHzE3pctQRJdNaRHNNcxGTFtb/
AFfk527fn8wDK7SW+9v2f9J1JPHfw+i0m1mvtw1DwbnS9F0eOM+TftepDazTXPziFU8xGQNBVLv8AmG4da+jwmGqyxNFVIVkc1zW0
dt0v9fij5GLULqXwLV02jktVfblPV/8AhLNO/wcfc9/8B2/wor9Tf+GGvjF/0Lun/wDgvj/+SKK+o+oYftP/AMC/
r+n6GX9o4nvD/
wAAj5f5f1pb+YbxH42kutQldSio44o4kjdZJkLxKz7VyXGwbndybPvNniptl7cXa3U3mlwtLcMqdC0hChEBz1bnnbkYzg1zl7of
mXcwZmVwSyApkS4+9L967NvGTHs59ufcfht4Pt737Abq184SzkZRRylsJdkMjntP+tdjBYDbsoA3b57B4enUhgPjPm30dlfTp8z1
qNecqsIvls3ba39bf1pb7P/AGZ/gzqur+E/DVwbaK031GSe4utSnGIIiWQRDOxYGMx5bY/
zebubKIUBb7rvPgXdajZy6Po13dGeKES1xb25e0nL8v2Z/
tC+Zt2uHwF7l6545X4CDTNF8J6do83miRLmSXTbcnNttm2+bDcwf8tli2ReUfMQjFJ8v0K/
Q3wbp9zFDBeSLHZM0YWIrgSW8gIwxit8J9mKgLu/eSmXcvK+X82VevPDlocig0mnacVJP4d1p/SR9dhYueGnFbyikvmo/
kfj7rHwE1PRjq0+ueI9e0m4CS2drBaieK09FwxWea4kjlk4svLiKRmMlvtDgSjg7vSPH8JfC/hi1m18QazP4/
intoIxY7rizktJLfwY5Z7iU3uUmUFEJ8wxN8yRX60Xngvw5d7bvUa+kAn2RRWksaxymbYs0xfz8bCq7Plydz9Mccc3w+svMHTT
NDsPrPMbgsNtIqkp507ZE3nFSJSOV2H0PvVccwrSai4UFfS6p3Pps76PT8WYTYkYUa/OLUXtH795N6aPTT290U+Y/
hT4T0Xwvq0V7DGkkrzlyrhB0oikwYYVmwPs4LAMAPN3fcTzZ9BaHPqVn4k+1aYtvBcLHcRq6WYhgjvQnmKkYfEwWqGMQzF9/
IUqQejsvh5otldTyhjBM4V/LMRCiYtwjxjecnDbyDwvQcHIPEtvs08EsBxI0rxKcZ+VSm5+G/2hx9ec4rb285e6+WzsvhXL/L/
Wlvdo4KjGpBrnnp7z/rp/
WwhHx2itdf0G0imjYy6ewe4hWQ4E5MJDN5FH7wxsdUpl2feb0R8veIvCwj+MobWDWdFhvitoy2uB5ckTRLGAYzKjY3fJn5SSEHXF
Yhpjns2uLSTgJUSiRYpLm9J27CocoiwZbcTlgf3q4440ePMNHsII5gGecgiXyxggF92QWzgLcMDDZz1G0fHqz9njqE3ooyTu/
RL+u53V8PGVKcYpttaLfqunU+M7D4Ig6lBdq+FYPtVrLILtc3WrK0ciEqAstsdy+wo+aXjfyxv67p37PNjqN5bX3izw3YNGjL
LHBpcwgWzlwSblliyYjys2qU040/3cc/
U2n+HtPjAdoLds28Fz5m052D5Q0ozzk8HArqHhMKRLHKmza4KrFtxjb38xsjrjheh45ruxGN9pQqQpySnKNo23v7u1tfl5LufPVM
koVpXrUfdovZuNkuXZ/1+J4Te+GdE0yykTtRe0UWlq++F1V47kRBFt4oVcxCB0BkEkg80yb00wbrU/
Mr9uSxiTwJpOrW8kd9Lp2qpfmKN+UMiJFJAGw2z0Qxk2n0wAxjg9V/FrWZSSKR4zM0oLUGHzd6xD9TECFZ2hvmTnDdACrY4/
Jz9s83F5p3hrS90wYtZ1dNUT44NgW5bSPsr+8bclXyrjaMKH84gPHzFHQnjKWZYao5txjVT1V09Laq+/m/
kdzoUcHg61KjTioFrTjYv2lpldPb7j8xrTSrya+ha1C7ZNPijUk48qVxd3IlwfvbCpTHy5zu3DG2v3F/
4Jk+DNN1TxrpXiTxPaNqs+l+Cn1jQLWkCwrafq0bXiC9F15dyZGRoInCeREtkrvAGT+Ndspgv7mNYmZLV7qCMK+0Mb0Ly00NrEfa
PtR03nyxBj955mV/cD/
gmXPawEmppPLP8AaofAH2e0McPnEvEdVluMguPLI8+FzcBcRu44Fp0VzeIqUfaW96SwZw6J6H5hmsVhKeKxFSHs4uT5vejD8nT5
n6z/wDC2fGv/P8A6j/4Mf8A7nor52/ti99NQ/8AAMf/AB6ivrPq1LTL7/6/p+lvhf7VxfeH/gC/z/q/
pb+YxfvDTxNdeWoMqWd1cRnZsErQeV+4L7j5ZkEnd/Pt2n5Dk4+0vgT8MLPwt08IQ+FrD9Q1y/
027v760ZGitiZJ4xbLrNLwMqLJakuN5TMxmHyR7Pm+ZPGXmWt7dW9xIIo4b97WMDk4BEKbyfkE5LZBVthQD58/L9a/
sKftc+FPA9i3wr+IHIDTNE1C/
wBc1G40TU5rS+1gWwSWX7PY20ZjZ44P9HW5ka9iDceBtgCEH4XL69BYfGSacvq8HKmk9XZxXzuv6Z+sYe1gVbU9JfEvtvtHy82f
ZfgHRJ9GmsDqUE1peRXU8E2nT25UQmBoxmG5L4kvt2M+RGrtXAYH5fuzwvc+c1rEzDHPF8rFvMEJiC8Kp2537wTym00tFJWt+Mv
BNudC1C/
wDFWmXmt+Itk1naR3lmZpLeeZorIm1FyZLaaQRu0iOdpjhc5jJb1zjwr8waXzjtEbLFcPmQuvD7mwm4gY2fKSM8HkdjgrYndV3f2
NRVpSycnyp6br5H22BhhY0nKnzOVk4tybV1y9PK3y/E+s9KmkUy7m2t5o12F2o8QkM/
l7+Ebvelbc88Pu3KeAvI4CIPPVobtwUwpJw+DGMn5R+AweDjka58y0vXTYfZbhLj95H5xJHR/
M8vGefl2bSP4s7vbL9/4sjELuXc3bsxhccgZxvJABxjKfXnpznKn8cfU7I+9JRdrPt2/
yX9WOk8ReIdK0qOR5GWW5x5Swb9jKXwMmTD7j9EHpx25LQob161CJ1c3YMc0qWca5um8zacW8QJ8/btG8/
IV+Tj5uPpp7051W6893lVwnbz1hTef3XLqW8z5c71AJ54PAWk4PxF8TvG3hDvYopDXgHxDqL380L3NjmJ7BU2hZPtRtpPs4m30d
uxvME0dy7MnuW69v+Z0KlCLTV7rzPetWsp5lla9iubeC1ac3VvLgYDXdJt82MH90yBG8pMyGqsFubBu+evE922k3FtqmnQXkAPd
SXCSS7kLke3aDzFmT8EwKhnTy2L0JgzckU5ta9+0f4i0Xsof+e0oj7XcrHLJHZPLFDfzykR7xf3kdvCFhbHy9im2G/
wA2T7nIr5fX4hffDx14rN5NbrJoWrn5LWIF4rGK1PCKnkxDdIs/zuFteYwSvy8erJepYimns2k+nSjvTV6kE9nJH2voerG/
0+1uVKEgFOV3Dcn3eD3yc8ZAxjvXSzX8cVuzSMFYidoLZyMHdzjOACD0Puk8A0aa7srcIrtFggMvJAZcg4GQBzngZ6fnoajrcxiU

PcHCrJk5+7kL2y0Dj88Z46VTowh0Mle6atdtrotV8v60tnXk1WhR05J6S01tZPfp8uxn+MdUIjvbtLgW/ZoLmYN5fnAIoTd/
FHg4PX24HGa+CPiJpZatG91qMsLWmtv000rXarbWcE0oUM1wMnLGMogj05vNy3+r2Yb6I8deLyLS9tLSV3d4ZEm2R+dvDzW8Ijk
jG3cJH0cnOzaAeo/Kr46/EgXPxF8Q+FIDQuLzwl4d05Nmu9KikikVZ9QvIVkkmhTzDILHhvj3Db5wJLg8AcFV3xLNWVnU6Lyj/
meHmlT2MJpW50XW//AG71+eh5XYyRTalqEUirmPVpvjngYXNjerG0nlTbXQEYkF3n96PLP2fy1/1wf5f18/YBu00XV/
B8t7cSW174zL8YaA2l25Ex0uG10sX9tdS3GV+1ecIZIkG8i1CbmcTPs2H8eFAl1kpmjBRvskUkEttbhtsM6hpiskI0TbtgtjzUPnEk
RgkFa/XL9lRtnKtPFXwkSaWXSbq/8AEWryNqrX0+KJZ9JmslJaAxxRH9tL7v0XftKbv+8v0eAxVhWoqDjZTVrxv/
AFsFA5LDC4jCV1JSu6btadv+fz+meJv+el5/35/+2UV1X/CvZf8AodrP/vtf/kiivrfr+I7x/wDAf6/p+h+ffUMP2n/4F/
X9P0P5QPfRrb3Ws+I0W5aXzvE2oX0UHXyFtIbua0zRRubIKo7bgQB02nApvwM+IUfw1+NPw+
+Js3h+Hxld+HNT1B5fCcixs3i6HUIIbc6LEz2d+9k7bA5vYr09MZ2g2jbgY8H4ovDe0yJPMG6e2DmMxNIsv1NIJmTc2zeLjaFLP
8AcJ3H0Az4U+0pPhT8XPhj8XYdEh8V3Pw58Rf25H4wvLlBky16P915tjc3bWmopAmKMp02nXuzLDyDuzXz+SYXBfV6lSkM6lWDb
UqjLfv3X8L7bn1MsXLCyXtG1NwtFTX3d0/8/8AM9j8aX8mqfEPV/
ibZeGhofim38Ww+Jd08I6je3tqlA01mvdRh8HPLIFWNYbdrpl1IabHgRY0l/OPL/
bP4X+LLLxP4A8E+PrK4sCnjzSk8QthYhLitQ5FubEXDSv9sazuYbmEzrDaFvvNbpuAH4V+Pvivrxx+I3jf4la9p9lPn9468XXvi
G30Kz2vD4at7qCa3td0tpkhtVv1sre4u42uBaaeJ/PU+RDsw335+wB46t9V8EaL8JtaFzG3wlg1HU/
BbTWxt059IvLxprrrTpZTJJsFvMsUsNwTJ5huHRYUAZYUYwrvE05+whyw0jyxUX9nqteP9Hluc1XGKU4811Zcqd9l/
X4bH6sabrKyxAMHfAUZYlQM5yVyrBs4Hpt00ua3Y74R0sgWB0KSRvDLKl2feFCtG/lv/
AKvDZULyWHOAK4CyuFuFCRYgwQ+WqOE2MzPHG8i43chCyKrLsPy2F7bvn/ZPKuSsUwhDk+YpYKcLjgE5B2/
XKjrXl1oRpv6cYXs2t9e3+Z9jgcVvqPsm4393aNT0r/mer+GbWzgs0MKyLerI0onjm01fNb03y/
KIIG3rvXjPHIrtUN3EsslwQ8UsEqzSm2eUmCfut5RK2feYrndu04jG018myeP/
AI1XM93F4J8JaTqFrbGNXmuNQisDiTf5BRZbKbfny5d2HG3C5+8K4vxH4n/
ad1a0Wz1S3/4RuAscX6DqdpM14uRvW5c28Xl+SNvkEK2fNL90d+SpU9yk1GpJ2g2rpPzXU9NzrzXJScVUlPbTJpPzue7N8GtA1/
VT4j1fSDd2tsb5p5LK5aKMM4MRiN3HNv8A+eLmBV6fvQcZwswq0C1tG+zadp9s1layXSxta28U2ibzNouH81vml+X+7I2EeXJ3N
fk974Y+MF88Nsde8W21vqRDT7vEKlK70Vz+5htwJ/OE7ZDNH5YTGsFYUNR8GeKycQ3XinXbFrRQIfsWot/
pPjJzWuwMCQx7E8sHG3z20ea5q2WYxP6xWqQaprmajFLS0ezf8AXyv6GFw0jT9vWq0pKn78LGCv9tFZ97/APDHuGu3Nrp0/
wBkMkLD2mbe8xk8x5AJAcHbnJw3fuTXmt7d3M7ypDGGjXHexL2DDZC7fK0c7W6Y6cjmUUtPCLKYbnVdabv6Sikxefu3jOPM3uZ2
3ltsfJUY2nrv6ld29rGiGRmysm1BLQY0C+Y8kpnZBdoK7QV07ceQ0vN0oqUHUltdVnkY/
EYVYoyg1zRV17t+kenU4C5vbXtTcSW4e0hsORPqmsPcMjBDpml7JrmdRswRF5qqylk3eYh3HZivx2j+Lumi/ET4neJtI0/
Sdbh8W+JvGY06710xWw0LS9WXTTrfT7i1VvM3G0Frcs0U8zzVwYvLy/
6iftkFfi3+Gpw58QDYbRtS8faPq2gQ56hFvLitGW1S7nsZDnORexGZV8vdiH940a/
HDRfMS3Lct5Wt4Yw4Xy1PzSOSIwX2E7yD8z22jky0KXlmtCuk3BSUm1stv8AgfPTU+WzbM4zw9eFWUXWlTahb3dfdtP57pddb
o+jvAizI9mHkM0rCSWSQL5cZkd97G0HL+Wh3YCB2AwOew/
Tn9mHWZL3xt4T0T2t105U1K7D5oJZ0mtYYUha1TKFSPTDGQq+SFTqTX5veALViyZSCdk0CY74PGc8Zx1I/Dmv0r/AGR9PL/
F74X3Hnqhuo/FntLb3iCRYiINMKicAvGJi3Lsuwqmgz7m5WvfwtdCXFFa61I9T4jnlP3ZNTNPN7L/8ij9H82P/Qbl/
wDAGf8A+S6K7T+2bn/og6R/34X/ABor6j2MP733/wDA/q/pbzfyw/vff/wP6v6W/kh1qK6eWRBoAtiGAYFV2Iz/
ALwsRHn5M5+bt505xnnj0Qkbbkx+V4Dwxwc+Z144xt2jJyDntXq+rwnI2ncb3B0WxguuBtXGcDaBnqQc9XBmF78hL03GGHf73PX8
OnevjcvqTw06d0k3yXtaWun9JHvNWhp1HKtLm54K6tJpXtDdt7h5YwWgzHhLXc0GUGxgSKecbQxz1zn7wxX3F+wjPJH8f/
EHgeS4mls/GXw0mktWuZizPq+juZLe1s/3WYBdi+mNw4dsC3i0yTAx8IaXdAKGMW791LFt3d5tgDfdxLPLZjGW3csMc/
Xf7JWu3+h/tT/s8X1ibZV1DxTLo+rLcxIq3GmzaVeCWNXJy/mKsSVf7in0Ye3hoRVzPx/f00/w27bImPvF0pJu91u32X+fu/
bw0lm0u6/
s3VXgju9sMzfZzhcQRmWgigXzAiB3BhLeHAGMYBGA6K5uVZVjWtZiz17buhHGTxnvnoe0c1btPds0pwW1vFRRBkvYYLwyB3eV
NV1JlllchMpcCREij3YhEDESSCTcef3U114e1qTR9djk3tC0UrBnimTGcocKdnKlvm+Xcv3s14141MZU9r8NKWlTLJcu9t/
1PtcLmNej0Ebx5Lq94q9LzB/I9DsLwIF8yFDFb8RqTIN2/
r80bR42hBjTcYIC4wd2Rqs32lp3t7bcGUuy9pofykjySxJnGednGafLpVwv0MttqNpPKs09I9mdjBmx8+0hdqk78HgfdaA0c89hp
3g+w1NdPuJLuSL7f9pjFvbgNKEj8nzVuMyLTLiRRGSDyr4zjFdNSpQjCTpyaqJXi+br7vb1f9NH1mFxcK0L83vaWa01uunfTtY+c
77XrqKCSyiu3ima4igLwVpNmI8pWiIz5tu0yLTYHoTwtJGbbDX1xei08a5t74K73VnPG0ux0sJWUmPcJfnDDy12BV5b0R9ZX+i
eHfdQdT0synlTJIGu2W42TnaFNjG1QuSCXXOSQoLdaDXC3g0vzEudQhtrZU55VtEmAW0WVJhhd+BOE+ZGLgEt5vmlnCuG8+piKz
pyTm2rbN6dP8v60t6P1zeUqUoxn7ttnr209NEeBzLqE7PtB0WZY0PGnNu546kg4wMY5JPI8+8R3k8cM1p5ypJ0j2bnaCI7ibELvG
8hdfJhkeF5xkUipamVYnj1nxdrtpoS3kCRKJo5Z0Ewl+SLGzGz92fMLL+RuXaFHx8vxd8UdRvW8N+NtSu7m6ijm8K699leC4MD
PMI4AJUJfaxSWdcvLMASPNbGM15deTq0Z05/DNwdtHa62fQ8fE1ZTTqSs5RStppuuH+bP7SvxVvfjN8SLj7DqkP8cFD8SeF9IsGs/
L3eIbcxR+KyJ/
OXz4N1rpBguPJkyhmEkk5t4c5v0mC8hTsZcrnaE3cAFhn0B457c4ri9FtrhtJtHnneWaYyyyyFTmUSSsVKlXcd9w2D9ouN26f5
MquwA+veFbU4SepwpJH97Pv6+v5jg162V040sPyRvaUb0+r2js3rc/
P8AMqcatdvJ0XNzJ2Uml9np8z60+H1rKbVDH5mUDtzjaLFPHQ5Eu/
r1XZzv3fL+jH70nh+Zvin80L5oJha2y3dxJPE5RLAXMMUbK67SjV027h14tohi58zK/
Fnw80Mw6fu+0bVaEhj5AZ1WTJLLH5w8zy9vzAFQdw+7X6u/speHhDYavr95uulKwx0qx0e4NttkkkjGp/bpNu+Qw+VvtPk3N5hc/
Mu3LdOS1Z4nMVSqW5YVUo8qs9+rPKxn+z4atWp6TpwvG+qvdLVddz7C/4Rbs/+g+f/AABP/wAnUu3Fp6z/APfo/
wCNf0x1al2l9/9f0/S3x/9q4vvd/wBF5/1f0t/LDqN3lPuXB5x8v0zj5vb/GvH9ZsSfN2nAz/AHSe/b5v519Bat/q/wAH/
wDQVrxrWpuy/wDAv51+U4etP21PVfEuh9tjYqrRq0fZbadYr9Dkd0jcfISpVbju/wBwj7b759c19M/
AG3yFHT4GajudTzeJr5sjKxpmx+e+SbkJtLhLbQltpXi5r5103/j4i+kv/stfu/wP/wCRx8E/9hr/AnoXFe5Tr1YtjJSV019leS/
z+/yKymhTvSWtubv2UT9/fA8i3eh6LHLG6LaLeW8+2feL1or+eW3vEmL5Wf0cGLZKMAHz0cDpFukW/
iSyW21LypLm13f2fd+WFlhjz5k5b4m5b5y4drLkC7DlWDHHLFDH/kTPDP/Xvd/wDpQa7HWP8AWQ/V/
wD0BK8TF1Z060uW37347rvy7dj6b6tS/vf+BP8Ar/
hz5x1bSvEwh6tZahps9yx01blTrCGUw294JRBk3AAkVvK8oeWCj48x+Rk1rwwFua0W2g8Q2l3YXKL00Uum037jb5fm+eohUSbz5e
w7A6C4k4bSvbkXv/H2f+B/0rxLxJ/yEr3/rLJ/WuZVJLr/Wn+X9aw78E/Y1accYX5eZan3vp/
wAA607+PPhK0K8xDq2o3Xyo8szsGQpv2LFML43Z075hjaOTXjetfFKLXni1T7q6hijuLnuIJYmiF92f3KRTbs/NmXc3L/
JheDnA8zf/AI+bv/ro/wD0Qa/4+h9R/Om6s2mnaz8j2q9eao1GrXUdNPT/
AIP3+R3V7f3mquwuGaS1cyzRox30r0I9yvLgF921cHYoXBwDuIHZD+0zrD6N8HVetY2paQS6R9h37CE1qa3gE5uNjH90LbeIjbn

```
zSceZGBk/Uft/qU+g/kK+Pf2yv8AkiXjP/r98P8A/pZJWdr2T2bX5nhYjGV/Y1NY/D/
Ku6Pz1t7JLS1ggVhJHbCOyWUJs80xQwzNKU3PsLfaQmzc+BHned2B6n4ThEjRAL90BsdckZ+nPHUn068585b/AFQ/6/H/
APSLTq9R8E/6wf8AXL/2Y17uJisFh6EqGjmlzcz5ui27Hzs37aSc+620tt/kj7N+HRsayS48poJLRVEbKpufM3n54jb/
ALrf5uwYbedhXLXzx+137PsVrD4S0iwsrdJH1G1huQkqiBYJwrmZHUh9hbevJILEYxxX4yfb7/X6Z/19xf8AobV+0Hws+7H/ALi/
+gy1nkEuXN8KraVq8VU13um902p5+bUoxy3GSV01SbXXW6PoX+yL7/nws/8AwIT/AON0VYor9c9lT/lf/gT/AK7/ANLX8u5p/
wA34Ly/4P3+R//Z"
```

7.4 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request
Content-Type	Returns the type of received request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

7.5 Response example:

Click to expand...

```
{
  "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"
}
]
}
]
}

```

7.6 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

Parameter	Parameter description
ProcessedPersons	Number of generated vectors
ProcessedPersonsDuration	Time spent generating vectors
PersonList	A list of users with information from the faces database (see Getting information about users).
id	User identifier from the faces database
Name	Name
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	true 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}` - to get an image of a user from the face database by their ID in *Axxon PSIM*.

8.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	ID of the Face recognition server in <i>Face PSIM</i>
id	Yes	the image ID from the reference face database in GUID format

8.3 Request example:

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

8.4 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

8.5 Response example:

JPEG image

9 Getting analytic data

9.1 General request format:

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

9.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	Identifier of the Face recognition server in <i>Face PSIM</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: 0 – Unknown, 1 – Males, 2 – Females
faceMode	No	Type of persons whose data should be obtained: 0 – all faces, 1 – only recognized faces, 2 – 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 PSIM</i>

9.3 Request example:

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

```
{
  "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 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request
Content-Type	Returns the type of received request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

9.5 Response **example**:

Click to expand...

```
{
  "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"
}
}

```

9.6 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 Getting the recognition protocol)
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 Getting information about users)
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	true if the unrecognized face was added automatically. false 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

Parameter	Parameter description
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
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
ProcessedTotal	Total number of processed faces
ProcessedDuration	Execution time of the faces processing in milliseconds
CompareCount	Total number of the face comparisons
Response	Response

Parameter	Parameter description
Status	Completed request status: SUCCESS — successful, if error — see List of possible errors of the Face recognition server

10 Getting analytic data by time

10.1 General request format:

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

10.2 Request parameters:

Parameter	Required	Parameter description
server_id	Yes	Identifier of the Face recognition server in <i>Face PSIM</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
cameras	No	Camera IDs in <i>Face PSIM</i> If not specified, then results for all cameras are displayed
genders	Yes	Gender of persons whose data should be obtained: 0 – Unknown, 1 – Males, 2 – Females If parameter value is empty, then all results are displayed regardless of gender
returnLocalTime	No	Time format of the face capture: 0 – time in UTC+0 format, 1 – local time of the Server If not specified, then returnLocalTime=0
timePeriod	No	Type of grouping the analytic data: 0 – by hour, 1 – by day, 2 – by month, 3 – by year If not specified, then timePeriod=0

10.3 Request example:

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

```

{
  "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 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request
Content-Type	Returns the type of received request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

10.5 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"  
}
```

10.6 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 Getting the recognition protocol)
Response	Response
Status	Completed request status: SUCCESS — successful, if error — see List of possible errors of the Face recognition server

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 Request headlines:

Headline	Headline description
Allow	Returns the HTTP method used in request
Content-Length	Shows the size of characters in bytes, received in the request
Content-Type	Returns the type of received request
Server	Specifies the server from which the request comes
Access-Control-Request-Method	Returns the HTTP method that can be used in the request
Access-Control-Origin	Indicates if the code that sends the request from this source can access the server response
Date	Shows the time of the request

11.5 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