

Integration Services

With Integration Services you can get more information on a subject and take an informed decision. We provide you with our own integration service, if you choose to use it, or you could create your own private integration service to perform specific analysis for your scenarios.

Integration Service Requirements

To create your own private integration service you need to comply with the following requirements.

Communication Requirements

A secure connection between Positive ID and your integration REST API.

- An HTTPS **POST** request through any valid HTTP port you deem necessary.
- Example: `https://rest.positiveidcheck.com:1250/integration`

Request Requirements

The integration service must expose an HTTP POST request expecting a multipart/form-data body.

Body

The body of the request must be a multipart/form-data containing a JSON object and/or an optional image. You must enable sending images in the app for Positive ID to send images to your integration service.

See form data description below:

Name	Type	Format	Required	Description
image	bytes	bytes	NO	The bytes of the image
metadata	string	JSON	YES	Mandatory JSON object/string with data described in section "JSON"

Image (form-data part)

The integration service must parse the form data checking if the image field is present and process accordingly.

See an example of an image inside the form data below:

```
Content-Type: image/jpeg

image raw bytes
```

JSON (form-data part)

The field metadata is a mandatory JSON string containing data as described below:

Name	Type	Format	Description
identifier	string	alphanumeric or null	The identifier assigned to the ID holder by the issuing authority.
name	string	alpha or null	The name of the ID holder. Content might vary per country.
issueDate	integer	Seconds since January 1st, 1970 or -1	The date the ID was issued by the issuing authority.
birthDate	integer	Seconds since January 1st, 1970 or -1	The date of birth of the ID holder.
expirationDate	integer	Seconds since January 1st, 1970 or -1	The date in which the ID is no longer valid.
sex	string	"M" = Male, "F" = Female, "U" = Unknown/Empty	The gender of the ID holder.
state	string	alpha or null	The state where the ID was issued.
country	string	ISO 3166-1 alpha-3 or null	The 3 letter code of the issuing country. Ex. USA, DEU, JPN.

See examples of a JSON string part inside the form data below:

Content-Type: application/json

```
{
  "identifier": "1234567",
  "name": "John Smith",
  "issueDate": 12312312312,
  "birthDate": 12312312312,
  "expirationDate": 123123123123,
  "sex": "M",
  "state": "Virginia",
  "country": "USA"
}
```

Content-Type: application/json

```
{
  "identifier": null,
  "name": "Jane Doe",
  "issueDate": -1,
}
```

```
"birthDate": 12312312312,  
"expirationDate": 123123123123,  
"sex": "F",  
"state": "Munchen",  
"country": "DEU"  
}
```

Form data

The previous examples described how an image and JSON look like inside a form data. The following is an example of how the form data must look including the image and JSON string parts:

```
Content-Type: multipart/form-data; boundary=123456789  
  
--123456789  
Content-Disposition: form-data; name="image"; filename="filename.jpeg"  
Content-Type: image/jpeg  
  
image raw bytes  
  
--123456789  
Content-Disposition: form-data; name="metadata"  
Content-Type: application/json  
  
{  
  "identifier": null,  
  "name": "Jane Doe",  
  "issueDate": -1,  
  "birthDate": 12312312312,  
  "expirationDate": 123123123123,  
  "sex": "F",  
  "state": "Munchen",  
  "country": "DEU"  
}  
  
--123456789--
```

Response

Errors

Return appropriate http status codes when an error is encountered:

400 Bad Request

1. Empty metadata field
2. Invalid JSON

3. Metadata field doesn't contain JSON object

500 Internal Server Error

1. Any internal error in the integration service.

Success

Successful requests to the integration service must return "200 - OK" status regardless if the data provider produces an error. These errors are reported inside the JSON string response.

Your integration service must respond with a JSON object as described below:

Name	Type	Format	Description
result	integer	0, 1, or -1	0 means red flag, 1 means green flag, -1 means an error occurred. The resultReason field must contain the appropriate message based on the result.
resultReason	string	alphanumeric	The reason for the result.
moreInfo	string	http(s) URL or null	An HTTP URL link where user could learn more information.
source	string	alphanumeric or null	System where the information came from.

Response Examples:

```
Content-Type: application/json
```

```
{
  "result": 1,
  "resultReason": "Subject seems clear.",
  "moreInfo": "https://positiveidcheck/moreinfo",
  "source": null
}
```

```
Content-Type: application/json
```

```
{
  "result": 0,
  "resultReason": "Subject has been previously convicted",
  "moreInfo": "https://positiveidcheck/moreinfo",
  "source": "FelonOnDemand"
}
```

```
Content-Type: application/json
```

```
{  
  "result": -1,  
  "resultReason": "Data provider is unreachable.",  
  "moreInfo": null,  
  "source": null  
}
```