

## Passport Box API

Generated by Doxygen 1.8.13

## Contents

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>Data Structure Index</b>	<b>2</b>
2.1	Data Structures . . . . .	2
<b>3</b>	<b>File Index</b>	<b>2</b>
3.1	File List . . . . .	2
<b>4</b>	<b>Data Structure Documentation</b>	<b>2</b>
4.1	PBAPI_AutoCaptureSettings Struct Reference . . . . .	2
4.1.1	Detailed Description . . . . .	3
4.2	PBAPI_CaptureSettings Struct Reference . . . . .	3
4.2.1	Detailed Description . . . . .	3
4.3	PBAPI_DocInfo Struct Reference . . . . .	3
4.3.1	Detailed Description . . . . .	4
4.4	PBAPI_ErrorInfo Struct Reference . . . . .	4
4.4.1	Detailed Description . . . . .	4
4.5	PBAPI_FieldInfo Struct Reference . . . . .	4
4.5.1	Detailed Description . . . . .	5
<b>5</b>	<b>File Documentation</b>	<b>5</b>
5.1	passport_box_api.h File Reference . . . . .	5
5.1.1	Detailed Description . . . . .	7
5.1.2	Function Documentation . . . . .	7
<b>6</b>	<b>Example Documentation</b>	<b>17</b>
6.1	passport_box_api_sample.cpp . . . . .	17
	<b>Index</b>	<b>21</b>

## 1 Overview

The Smart Passport Box API Library allows to recognize Internal passport of Russia using Smart Passport Box device.

This file contains a brief description of all members of the Library. Sample usage is shown in the `passport_box_api_sample.cpp`.

Feel free to send any questions about the Library on [support@smartengines.biz](mailto:support@smartengines.biz).

## 2 Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<b>PBAPI_AutoCaptureSettings</b>	
The structure represents settings for auto capture mode	<b>2</b>
<b>PBAPI_CaptureSettings</b>	
The structure represents the capture document settings	<b>3</b>
<b>PBAPI_DocInfo</b>	
The structure represents document info	<b>3</b>
<b>PBAPI_ErrorInfo</b>	
The structure represents last error description	<b>4</b>
<b>PBAPI_FieldInfo</b>	
The structure represents field info	<b>4</b>

## 3 File Index

### 3.1 File List

Here is a list of all documented files with brief descriptions:

<b><a href="#">passport_box_api.h</a></b>	
Passport Box API Include File	<b>5</b>

## 4 Data Structure Documentation

### 4.1 PBAPI\_AutoCaptureSettings Struct Reference

The structure represents settings for auto capture mode.

```
#include <passport_box_api.h>
```

### Data Fields

- int [interval](#)  
*Time interval (in milliseconds) to check the document on the working surface.*
- double [min\\_score](#)  
*The minimum required detection score to start recognition process (from 0.0 to 1.0)*

#### 4.1.1 Detailed Description

The structure represents settings for auto capture mode.

Examples:

[passport\\_box\\_api\\_sample.cpp](#).

## 4.2 PBAPI\_CaptureSettings Struct Reference

The structure represents the capture document settings.

```
#include <passport_box_api.h>
```

### Data Fields

- int [timeout](#)  
*Specifies maximum recognition time in milliseconds.*
- char [doc\\_classes](#) [4096]  
*Specifies the list of document classes for recognition separated via semicolon.*
- bool [compete\\_doc\\_classes](#)  
*Specifies whether the documents should be compete in recognition process.*

#### 4.2.1 Detailed Description

The structure represents the capture document settings.

Examples:

[passport\\_box\\_api\\_sample.cpp](#).

## 4.3 PBAPI\_DocInfo Struct Reference

The structure represents document info.

```
#include <passport_box_api.h>
```

#### Data Fields

- char [doc\\_type](#) [256]  
*Type of recognized document(empty if document wasn't recognized)*
- int [field\\_count](#)  
*Number of text fields in recognized document.*
- int [image\\_count](#)  
*Number of image fields in recognized document.*

#### 4.3.1 Detailed Description

The structure represents document info.

Examples:

[passport\\_box\\_api\\_sample.cpp](#).

### 4.4 PBAPI\_ErrorInfo Struct Reference

The structure represents last error description.

```
#include <passport_box_api.h>
```

#### Data Fields

- int [err\\_code](#)  
*Error code.*
- char [err\\_message](#) [1024]  
*Error description.*

#### 4.4.1 Detailed Description

The structure represents last error description.

Examples:

[passport\\_box\\_api\\_sample.cpp](#).

### 4.5 PBAPI\_FieldInfo Struct Reference

The structure represents field info.

```
#include <passport_box_api.h>
```

### Data Fields

- char [field\\_name](#) [256]  
*Name of text field.*
- char [field\\_value](#) [4096]  
*Value of text field.*
- bool [is\\_accepted](#)  
*Bool flag tells is system sure in result.*

#### 4.5.1 Detailed Description

The structure represents field info.

Examples:

[passport\\_box\\_api\\_sample.cpp](#).

## 5 File Documentation

### 5.1 passport\_box\_api.h File Reference

Passport Box API Include File.

#### Data Structures

- struct [PBAPI\\_ErrorInfo](#)  
*The structure represents last error description.*
- struct [PBAPI\\_DocInfo](#)  
*The structure represents document info.*
- struct [PBAPI\\_FieldInfo](#)  
*The structure represents field info.*
- struct [PBAPI\\_CaptureSettings](#)  
*The structure represents the capture document settings.*
- struct [PBAPI\\_AutoCaptureSettings](#)  
*The structure represents settings for auto capture mode.*

#### Macros

- `#define PBAPI\_NO\_ERROR 0`  
*No error has occurred. It indicates that the function completed successfully.*
- `#define PBAPI\_BAD\_ARGS -1`  
*This error indicates that one or more arguments passed to the function are not correct.*
- `#define PBAPI\_INTERNAL\_ERROR -2`  
*An internal error has occurred. This error indicates that something went wrong.*
- `#define PBAPI\_CAPTURE\_ERROR -3`  
*A capture error has occurred. This error indicates problems with capture device.*
- `#define PBAPI\_ENGINE\_ERROR -4`  
*A common recognition engine problem (not configured, not loaded and etc.)*

## Typedefs

- typedef void(\* [PBAPI\\_CaptureDocumentCB](#)) (int, void \*cb\_data)  
*Specifies the signature of callback recognition function.*

## Functions

- int [PBAPI\\_EXPORT PBAPI\\_Configure](#) (const char \*config\_path)  
*Configures the recognition engines.*
- int [PBAPI\\_EXPORT PBAPI\\_GetDocumentClasses](#) (char \*doc\_classes\_buf, int doc\_classes\_buf\_size)  
*Returns the list of supported document classes as a text string separated via semicolon.*
- int [PBAPI\\_EXPORT PBAPI\\_OpenCaptureDevice](#) ()  
*Opens capture device.*
- int [PBAPI\\_EXPORT PBAPI\\_CloseCaptureDevice](#) ()  
*Closes capture device.*
- int [PBAPI\\_EXPORT PBAPI\\_GetLastError](#) (PBAPI\_ErrorInfo \*error)  
*Returns last error information.*
- int [PBAPI\\_EXPORT PBAPI\\_CaptureDocument](#) (const [PBAPI\\_CaptureSettings](#) \*capture\_settings=&PBAPI\_DEFAULT\_CAPTURESETTINGS)  
*Recognizes document on the working surface in synchronous mode.*
- int [PBAPI\\_EXPORT PBAPI\\_CaptureDocumentAsync](#) ([PBAPI\\_CaptureDocumentCB](#) cb, void \*cb\_data=0, const [PBAPI\\_CaptureSettings](#) \*capture\_settings=&PBAPI\_DEFAULT\_CAPTURESETTINGS)  
*Recognizes document on the working surface in asynchronous mode.*
- int [PBAPI\\_EXPORT PBAPI\\_StartAutoCaptureDocument](#) ([PBAPI\\_CaptureDocumentCB](#) cb, void \*cb\_data=0, const [PBAPI\\_AutoCaptureSettings](#) \*auto\_settings=&PBAPI\_DEFAULT\_AUTOCAPTURESETTINGS, const [PBAPI\\_CaptureSettings](#) \*capture\_settings=&PBAPI\_DEFAULT\_CAPTURESETTINGS)  
*Begins auto capture mode.*
- int [PBAPI\\_EXPORT PBAPI\\_StopAutoCaptureDocument](#) ()  
*Stops auto capture mode.*
- int [PBAPI\\_EXPORT PBAPI\\_RecognizeCurrentFrame](#) (bool new\_session=false, const [PBAPI\\_CaptureSettings](#) \*capture\_settings=&PBAPI\_DEFAULT\_CAPTURESETTINGS)  
*Process current frame from opened capture device.*
- int [PBAPI\\_EXPORT PBAPI\\_IsTerminal](#) (bool \*is\_terminal)  
*PBAPI\_IsTerminal Specifies whether the recognition should stop. It must be called together with PBAPI\_RecognizeCurrentFrame.*
- int [PBAPI\\_EXPORT PBAPI\\_GetDocumentInfo](#) (PBAPI\_DocInfo \*doc\_info)  
*Returns document info.*
- int [PBAPI\\_EXPORT PBAPI\\_GetFieldInfo](#) (int field\_index, [PBAPI\\_FieldInfo](#) \*field\_info)  
*Returns field info.*
- int [PBAPI\\_EXPORT PBAPI\\_GetImageName](#) (int image\_index, char \*image\_name, int size)  
*Returns image name.*
- int [PBAPI\\_EXPORT PBAPI\\_GetImageBase64Size](#) (int image\_index, int \*size)  
*Returns image size.*
- int [PBAPI\\_EXPORT PBAPI\\_GetImageBase64](#) (int image\_index, char \*data\_buffer, int size)  
*Returns image.*
- int [PBAPI\\_EXPORT PBAPI\\_SaveImage](#) (int image\_index, const char \*image\_path)  
*Saves image to specified path.*
- int [PBAPI\\_EXPORT PBAPI\\_TakeSnapshot](#) ()  
*Takes snapshot and saves it in internal buffer.*
- int [PBAPI\\_EXPORT PBAPI\\_GetSnapshotSize](#) (int \*size)  
*Returns snapshot size.*

- int PBAPI\_EXPORT [PBAPI\\_GetSnapshot](#) (char \*data\_buffer, int size)  
*Returns snapshot.*
- int PBAPI\_EXPORT [PBAPI\\_GetSnapshotBase64Size](#) (int \*size)  
*Returns snapshot size.*
- int PBAPI\_EXPORT [PBAPI\\_GetSnapshotBase64](#) (char \*data\_buffer, int size)  
*Returns snapshot.*
- int PBAPI\_EXPORT [PBAPI\\_GetSnapshotData](#) (PBAPI\_ImageData \*image\_data)  
*Returns snapshot in bitmap presence.*
- int PBAPI\_EXPORT [PBAPI\\_SaveSnapshot](#) (const char \*file\_path)  
*Takes snapshot and saves it to specified path.*

#### Variables

- const [PBAPI\\_AutoCaptureSettings](#) [PBAPI\\_DEFAULT\\_AUTOCAPTURESETTINGS](#) = { 1000, 0.5 }  
*Specifies default auto capture settings.*

#### 5.1.1 Detailed Description

Passport Box API Include File.

Copyright (c) 2012-2016, Smart Engines. All rights reserved.

#### 5.1.2 Function Documentation

##### 5.1.2.1 PBAPI\_CaptureDocument()

```
int PBAPI_EXPORT PBAPI_CaptureDocument (
    const PBAPI\_CaptureSettings * capture_settings = &PBAPI_DEFAULT_CAPTURESETTINGS )
```

Recognizes document on the working surface in synchronous mode.

#### Parameters

<i>capture_settings</i>	The pointer to capture settings
-------------------------	---------------------------------

#### Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

The function returns control when either document on the working surface is confidently recognized or break condition arises.

#### Examples:

[passport\\_box\\_api\\_sample.cpp](#).



### 5.1.2.2 PBAPI\_CaptureDocumentAsync()

```
int PBAPI_EXPORT PBAPI_CaptureDocumentAsync (
    PBAPI_CaptureDocumentCB cb,
    void * cb_data = 0,
    const PBAPI_CaptureSettings * capture_settings = &PBAPI_DEFAULT_CAPTURESETTINGS )
```

Recognizes document on the working surface in asynchronous mode.

#### Parameters

<i>cb</i>	The pointer to callback function
<i>cb_data</i>	The pointer to user defined callback parameter
<i>capture_settings</i>	The pointer to capture settings

#### Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

The function returns control and call callback function when either document on the working surface is confidently recognized or break condition arises.

#### Examples:

[passport\\_box\\_api\\_sample.cpp](#).

### 5.1.2.3 PBAPI\_CloseCaptureDevice()

```
int PBAPI_EXPORT PBAPI_CloseCaptureDevice ( )
```

Closes capture device.

#### Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

#### Examples:

[passport\\_box\\_api\\_sample.cpp](#).

### 5.1.2.4 PBAPI\_Configure()

```
int PBAPI_EXPORT PBAPI_Configure (
    const char * config_path )
```

Configures the recognition engines.

## Parameters

<i>config_path</i>	Configuration file path
--------------------	-------------------------

## Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

Configuration file contains the following options:

- Capture device name (should be changed only by experts)
- Capture frame size and FPS (should be changed only by experts)
- Key points for cropping capture frames (should be changed only by experts)
- Cropped frame size (should be changed only by experts)
- Engine configurations (should be changed only by experts)

## Examples:

[passport\\_box\\_api\\_sample.cpp](#).

## 5.1.2.5 PBAPI\_GetDocumentClasses()

```
int PBAPI_EXPORT PBAPI_GetDocumentClasses (
    char * doc_classes_buf,
    int doc_classes_buf_size )
```

Returns the list of supported document classes as a text string separated via semicolon.

## Parameters

<i>doc_classes_buf</i>	Buffer for document classes
<i>doc_classes_buf_size</i>	Size of passed buffer

## Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

## Examples:

[passport\\_box\\_api\\_sample.cpp](#).

## 5.1.2.6 PBAPI\_GetDocumentInfo()

```
int PBAPI_EXPORT PBAPI_GetDocumentInfo (
    PBAPI_DocInfo * doc_info )
```

Returns document info.

**Parameters**

<i>doc_info</i>	The pointer to document info structure
-----------------	--

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**Examples:**

[passport\\_box\\_api\\_sample.cpp](#).

**5.1.2.7 PBAPI\_GetFieldInfo()**

```
int PBAPI_EXPORT PBAPI_GetFieldInfo (
    int field_index,
    PBAPI_FieldInfo * field_info )
```

Returns field info.

**Parameters**

<i>field_info</i>	The pointer to field info structure
<i>field_index</i>	Index number of field in doc

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**Examples:**

[passport\\_box\\_api\\_sample.cpp](#).

**5.1.2.8 PBAPI\_GetImageBase64()**

```
int PBAPI_EXPORT PBAPI_GetImageBase64 (
    int image_index,
    char * data_buffer,
    int size )
```

Returns image.

**Parameters**

<i>image_index</i>	The inside index (should be from 0 to image_count-1) of image
<i>data_buffer</i>	The pointer to data buffer with image in base64 (data buffer should be preallocated)
<i>size</i>	Size of preallocated data_buffer

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**5.1.2.9 PBAPI\_GetImageBase64Size()**

```
int PBAPI_EXPORT PBAPI_GetImageBase64Size (
    int image_index,
    int * size )
```

Returns image size.

**Parameters**

<i>image_index</i>	The inside index (should be from 0 to image_count-1) of image
<i>size</i>	Demanded size of buffer

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**Examples:**

[passport\\_box\\_api\\_sample.cpp](#).

**5.1.2.10 PBAPI\_GetImageName()**

```
int PBAPI_EXPORT PBAPI_GetImageName (
    int image_index,
    char * image_name,
    int size )
```

Returns image name.

**Parameters**

<i>image_index</i>	The inside index (should be from 0 to image_count-1) of image
<i>image_name</i>	The pointer to string buffer for image name
<i>size</i>	Size of preallocated image_name

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**Examples:**

[passport\\_box\\_api\\_sample.cpp](#).

#### 5.1.2.11 PBAPI\_GetLastError()

```
int PBAPI_EXPORT PBAPI_GetLastError (
    PBAPI_ErrorInfo * error )
```

Returns last error information.

##### Parameters

<i>error</i>	The pointer to error description structure
--------------	--

##### Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

##### Examples:

[passport\\_box\\_api\\_sample.cpp](#).

#### 5.1.2.12 PBAPI\_GetSnapshot()

```
int PBAPI_EXPORT PBAPI_GetSnapshot (
    char * data_buffer,
    int size )
```

Returns snapshot.

##### Parameters

<i>data_buffer</i>	The pointer to data buffer with image (data buffer should be preallocated)
<i>size</i>	Size of preallocated data_buffer

##### Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

#### 5.1.2.13 PBAPI\_GetSnapshotBase64()

```
int PBAPI_EXPORT PBAPI_GetSnapshotBase64 (
    char * data_buffer,
    int size )
```

Returns snapshot.

##### Parameters

<i>data_buffer</i>	The pointer to data buffer with image in base64 (data buffer should be preallocated)
<i>size</i>	Size of preallocated data_buffer

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**5.1.2.14 PBAPI\_GetSnapshotBase64Size()**

```
int PBAPI_EXPORT PBAPI_GetSnapshotBase64Size (
    int * size )
```

Returns snapshot size.

**Parameters**

<i>size</i>	Demanded size of buffer
-------------	-------------------------

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**5.1.2.15 PBAPI\_GetSnapshotData()**

```
int PBAPI_EXPORT PBAPI_GetSnapshotData (
    PBAPI_ImageData * image_data )
```

Returns snapshot in bitmap presence.

**Parameters**

<i>image_data</i>	The pointer to PBAPI_ImageData structure
-------------------	--

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**5.1.2.16 PBAPI\_GetSnapshotSize()**

```
int PBAPI_EXPORT PBAPI_GetSnapshotSize (
    int * size )
```

Returns snapshot size.

**Parameters**

<i>size</i>	Demanded size of buffer
-------------	-------------------------

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**5.1.2.17 PBAPI\_IsTerminal()**

```
int PBAPI_EXPORT PBAPI_IsTerminal (
    bool * is_terminal )
```

PBAPI\_IsTerminal Specifies whether the recognition should stop. It must be called together with PBAPI\_↔RecognizeCurrentFrame.

**Parameters**

<i>is_terminal</i>	The pointer to is_terminal flag
--------------------	---------------------------------

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**5.1.2.18 PBAPI\_OpenCaptureDevice()**

```
int PBAPI_EXPORT PBAPI_OpenCaptureDevice ( )
```

Opens capture device.

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

The capture name to be open is specified in configuration file. Opening capture device for a long time reduces device life cycle.

**Examples:**

[passport\\_box\\_api\\_sample.cpp](#).

**5.1.2.19 PBAPI\_RecognizeCurrentFrame()**

```
int PBAPI_EXPORT PBAPI_RecognizeCurrentFrame (
    bool new_session = false,
    const PBAPI_CaptureSettings * capture_settings = &PBAPI_DEFAULT_CAPTURESETTINGS )
```

Process current frame from opened capture device.

## Parameters

<i>new_session</i>	Flag shows should recognition engine use old or new sessions(when sessions reset all previous results loses)
<i>capture_settings</i>	The pointer to capture settings

## Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

The function returns control when frame is recognized

## Examples:

[passport\\_box\\_api\\_sample.cpp](#).

## 5.1.2.20 PBAPI\_SaveImage()

```
int PBAPI_EXPORT PBAPI_SaveImage (
    int image_index,
    const char * image_path )
```

Saves image to specified path.

## Parameters

<i>image_index</i>	The inside index(should be from 0 to image_count-1) of image
<i>image_path</i>	Image file path

## Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

## 5.1.2.21 PBAPI\_SaveSnapshot()

```
int PBAPI_EXPORT PBAPI_SaveSnapshot (
    const char * file_path )
```

Takes snapshot and saves it to specified path.

## Parameters

<i>file_path</i>	Snapshot file path
------------------	--------------------



**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**Examples:**

[passport\\_box\\_api\\_sample.cpp](#).

**5.1.2.22 PBAPI\_StartAutoCaptureDocument()**

```
int PBAPI_EXPORT PBAPI_StartAutoCaptureDocument (
    PBAPI_CaptureDocumentCB cb,
    void * cb_data = 0,
    const PBAPI_AutoCaptureSettings * auto_settings = &PBAPI_DEFAULT_AUTOCAPTURESETTINGS,
    const PBAPI_CaptureSettings * capture_settings = &PBAPI_DEFAULT_CAPTURESETTINGS )
```

Begins auto capture mode.

**Parameters**

<i>cb</i>	The pointer to callback function
<i>cb_data</i>	The pointer to user defined callback parameter
<i>auto_settings</i>	The pointer to auto capture mode settings
<i>capture_settings</i>	The pointer to capture settings

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

In auto capture recognition mode the working surface is analyzed for the presence of the document. When the document is detected and recognized the function call specified callback function.

**Examples:**

[passport\\_box\\_api\\_sample.cpp](#).

**5.1.2.23 PBAPI\_StopAutoCaptureDocument()**

```
int PBAPI_EXPORT PBAPI_StopAutoCaptureDocument ( )
```

Stops auto capture mode.

**Returns**

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

**Examples:**

[passport\\_box\\_api\\_sample.cpp](#).

### 5.1.2.24 PBAPI\_TakeSnapshot()

```
int PBAPI_EXPORT PBAPI_TakeSnapshot ( )
```

Takes snapshot and saves it in internal buffer.

#### Returns

Returns error code. To obtain error description see [PBAPI\\_GetLastError\(\)](#) function

Use [PBAPI\\_GetSnapshotBase64\(\)](#) or [PBAPI\\_SaveSnapshot\(\)](#) to obtain saved snapshot.

#### Examples:

[passport\\_box\\_api\\_sample.cpp](#).

## 6 Example Documentation

### 6.1 passport\_box\_api\_sample.cpp

```
#ifndef _CRT_SECURE_NO_WARNINGS
#define _CRT_SECURE_NO_WARNINGS
#endif // _CRT_SECURE_NO_WARNINGS

#include <passport_box_api/passport_box_api.h>

#include <stdexcept>
#include <iostream>
#include <cstring>
#include <string>
#include <thread>

#ifdef WIN32
#include <windows.h>
#include <conio.h>
#endif // WIN32

static std::string utf8ToConsole(const std::string &utf8)
{
#ifdef WIN32

    int wide_size = MultiByteToWideChar(CP_UTF8, 0, utf8.c_str(), utf8.size(), 0, 0) + 1;
    wchar_t *wide = new wchar_t[wide_size]; memset(wide, 0, sizeof(wchar_t) * wide_size);
    MultiByteToWideChar(CP_UTF8, 0, utf8.c_str(), utf8.size(), wide, wide_size);

    int local_size = WideCharToMultiByte(CP_ACP, 0, wide, wide_size, 0, 0, 0, 0) + 1;
    char *local = new char[local_size]; memset(local, 0, sizeof(char) * local_size);
    WideCharToMultiByte(CP_ACP, 0, wide, wide_size, local, local_size, 0, 0);

    std::string cyrillic = local;

    delete[] local;
    delete[] wide;

#else // WIN32

    std::string cyrillic = utf8;

#endif // WIN32

    return cyrillic;
}

static void ConvertTo25Symbols(std::string& str) {
    while (str.size() < 25) {
        str.push_back(' ');
    }
}

static void printTextRecogResult()
```

```

{
    try
    {
        PBAPI_DocInfo docInfo = {0};
        PBAPI_GetDocumentInfo(&docInfo);
        std::cout << "Document: " << docInfo.doc_type << std::endl;
        std::cout << "Field count: " << docInfo.field_count << std::endl;
        std::cout << "Image count: " << docInfo.image_count << std::endl;

        for (int i = 0; i < docInfo.image_count; i++)
        {
            int image_size = 0;
            char image_name[4096] = { 0 };
            PBAPI_GetImageName(i, image_name, sizeof(image_name));
            PBAPI_GetImageBase64Size(i, &image_size);
            std::cout << "Image " << i << ", name: " << image_name << ", size (in bytes, base64): " << image_size
                << std::endl;
        }

        PBAPI_FieldInfo fieldInfo = {0};
        std::string tmpFieldName, tmpFieldValue, tmpFieldAccepted;
        for (int i = 0; i < docInfo.field_count; ++i)
        {
            PBAPI_GetFieldInfo(i, &fieldInfo);
            tmpFieldAccepted = fieldInfo.is_accepted ? " [+]" : " [-] ";
            tmpFieldName = fieldInfo.field_name;
            tmpFieldName += ' ';
            ConvertTo25Symbols(tmpFieldName);
            tmpFieldValue = fieldInfo.field_value;
            std::cout << tmpFieldName << tmpFieldAccepted << utf8ToConsole(tmpFieldValue) << std::endl;
        }
    }
    catch(std::exception &ex)
    {
        std::cerr << ex.what() << std::endl;
    }
}

static void documentCaptureCB(int ret_code, void *cb_data)
{
    std::cout << std::endl;

    if (ret_code == PBAPI_CB_DOCUMENT_DETECTED)
    {
        std::cout << "Document was detected" << std::endl;
    }
    else if (ret_code == PBAPI_CB_DOCUMENT_RECOGNIZED)
    {
        printTextRecogResult();
    }
    else if (ret_code == PBAPI_CB_DOCUMENT_REMOVED)
    {
        std::cout << "Document was removed" << std::endl;
    }
    else if (ret_code == PBAPI_CB_ERROR)
    {
        PBAPI_ErrorInfo error_info = { 0 }; PBAPI_GetLastError(&error_info);
        std::cerr << error_info.err_code << ", " << error_info.err_message << std::endl;
    }
}

int main(int argc, char *argv[])
{
    #if WIN32
        setlocale(LC_CTYPE, "Russian");
    #endif
    try
    {
        int snapshot_index = 0;
        if (PBAPI_Configure("passport_box_api.json") < 0)
            throw std::runtime_error("Something went wrong with configuration");

        if (PBAPI_OpenCaptureDevice() < 0)
            throw std::runtime_error("Couldn't open capture device");

        std::cout << "Smart PassportBox API Sample application" << std::endl;
        std::cout << "\tPress '0' to specify recognized documents" << std::endl;
        std::cout << "\tPress '1' to recognize passport in synchronous mode" << std::endl;
        std::cout << "\tPress '2' to recognize passport in asynchronous mode" << std::endl;
        std::cout << "\tPress '3' to start/stop auto recognition" << std::endl;
        std::cout << "\tPress '4' to recognize current frame" << std::endl;
        std::cout << "\tPress 's' to take snapshot" << std::endl;
        std::cout << "\tPress 'q' to exit" << std::endl << std::endl;

        bool is_auto_capture_enabled = false;
        bool should_exit = false;
    }
}

```

```

PBAPI_CaptureSettings capture_settings = PBAPI_DEFAULT_CAPTURESETTINGS;
PBAPI_AutoCaptureSettings auto_settings =
    PBAPI_DEFAULT_AUTOCAPTURESETTINGS;

while (should_exit == false)
{
    std::cout << "Press {0|1|2|3|4|s|q}: ";
#ifdef WIN32
    int c = _getch();
    std::cout << (char)c << std::endl;
#else // WIN32
    int c = getchar();
#endif // WIN32

    try
    {
        switch (c)
        {
            case '0':
            {
                char supported_doc_classes_buf[4096] = { 0 };
                PBAPI_GetDocumentClasses(supported_doc_classes_buf, sizeof(
                    supported_doc_classes_buf));

                std::cout << "Supported documents: " << supported_doc_classes_buf << std::endl;
                std::cout << "Enter a set of documents to be recognized (empty string - all documents): ";
                std::string doc_classes; std::getline(std::cin, doc_classes);

                memset(capture_settings.doc_classes, 0, sizeof(capture_settings.
                    doc_classes));
                strcpy(capture_settings.doc_classes, doc_classes.c_str());
                break;
            }
            case '1':
            {
                if (PBAPI_CaptureDocument(&capture_settings) < 0)
                    throw std::exception();

                printTextRecogResult();
                break;
            }
            case '2':
            {
                if (PBAPI_CaptureDocumentAsync(documentCaptureCB, 0, &capture_settings)
                    < 0)
                    throw std::exception();

                std::cout << "When the passport is recognized the callback function will print results" <<
                    std::endl;
                break;
            }
            case '3':
            {
                if (is_auto_capture_enabled)
                {
                    if (PBAPI_StopAutoCaptureDocument() < 0)
                        throw std::exception();

                    is_auto_capture_enabled = false;
                    std::cout << "Auto capture mode was disabled" << std::endl;
                }
                else
                {
                    if (PBAPI_StartAutoCaptureDocument(documentCaptureCB, 0, &
                        auto_settings, &capture_settings) < 0)
                        throw std::exception();

                    is_auto_capture_enabled = true;
                    std::cout << "Auto capture mode was enabled. Put the document on workspace and wait..." <<
                        std::endl;
                }
                break;
            }
            case '4':
            {
                if (PBAPI_RecognizeCurrentFrame(true, &capture_settings) < 0)
                    throw std::exception();

                printTextRecogResult();
                break;
            }
            case 's':
            {
                char snapshot_name[1024] = { 0 };

```

```

    sprintf(snapshot_name, "%06d.jpg", snapshot_index++);
    if (PBAPI_TakeSnapshot() < 0)
        throw std::exception();
    if (PBAPI_SaveSnapshot(snapshot_name) < 0)
        throw std::exception();

    std::cout << "Snapshot saved to " << snapshot_name << std::endl;
    break;
}

case 'q':
{
    should_exit = true;
    break;
}
}
catch (...)
{
    PBAPI_ErrorInfo error_info = { 0 }; PBAPI_GetLastError(&error_info);
};
std::cerr << error_info.err_code << ", " << error_info.err_message
err_message << std::endl;
}
}
}
catch (...)
{
    PBAPI_ErrorInfo error_info = { 0 }; PBAPI_GetLastError(&error_info);
    std::cerr << error_info.err_code << ", " << error_info.err_message << std::endl;
}

if (PBAPI_StopAutoCaptureDocument() < 0)
{
    PBAPI_ErrorInfo error_info = { 0 }; PBAPI_GetLastError(&error_info);
    std::cerr << error_info.err_code << ", " << error_info.err_message << std::endl;
}

if (PBAPI_CloseCaptureDevice() < 0)
{
    PBAPI_ErrorInfo error_info = { 0 }; PBAPI_GetLastError(&error_info);
    std::cerr << error_info.err_code << ", " << error_info.err_message << std::endl;
}

return 0;
}

```

## Index

PBAPL\_AutoCaptureSettings, [2](#)  
PBAPL\_CaptureDocument  
    passport\_box\_api.h, [7](#)  
PBAPL\_CaptureDocumentAsync  
    passport\_box\_api.h, [7](#)  
PBAPL\_CaptureSettings, [3](#)  
PBAPL\_CloseCaptureDevice  
    passport\_box\_api.h, [8](#)  
PBAPL\_Configure  
    passport\_box\_api.h, [8](#)  
PBAPL\_DocInfo, [3](#)  
PBAPL\_ErrorInfo, [4](#)  
PBAPL\_FieldInfo, [4](#)  
PBAPL\_GetDocumentClasses  
    passport\_box\_api.h, [9](#)  
PBAPL\_GetDocumentInfo  
    passport\_box\_api.h, [9](#)  
PBAPL\_GetFieldInfo  
    passport\_box\_api.h, [10](#)  
PBAPL\_GetImageBase64  
    passport\_box\_api.h, [10](#)  
PBAPL\_GetImageBase64Size  
    passport\_box\_api.h, [11](#)  
PBAPL\_GetImageName  
    passport\_box\_api.h, [11](#)  
PBAPL\_GetLastError  
    passport\_box\_api.h, [11](#)  
PBAPL\_GetSnapshot  
    passport\_box\_api.h, [12](#)  
PBAPL\_GetSnapshotBase64  
    passport\_box\_api.h, [12](#)  
PBAPL\_GetSnapshotBase64Size  
    passport\_box\_api.h, [13](#)  
PBAPL\_GetSnapshotData  
    passport\_box\_api.h, [13](#)  
PBAPL\_GetSnapshotSize  
    passport\_box\_api.h, [13](#)  
PBAPL\_IsTerminal  
    passport\_box\_api.h, [14](#)  
PBAPL\_OpenCaptureDevice  
    passport\_box\_api.h, [14](#)  
PBAPL\_RecognizeCurrentFrame  
    passport\_box\_api.h, [14](#)  
PBAPL\_SaveImage  
    passport\_box\_api.h, [15](#)  
PBAPL\_SaveSnapshot  
    passport\_box\_api.h, [15](#)  
PBAPL\_StartAutoCaptureDocument  
    passport\_box\_api.h, [16](#)  
PBAPL\_StopAutoCaptureDocument  
    passport\_box\_api.h, [16](#)  
PBAPL\_TakeSnapshot  
    passport\_box\_api.h, [16](#)  
passport\_box\_api.h, [5](#)  
    PBAPL\_CaptureDocument, [7](#)  
PBAPL\_CaptureDocumentAsync, [7](#)  
PBAPL\_CloseCaptureDevice, [8](#)  
PBAPL\_Configure, [8](#)  
PBAPL\_GetDocumentClasses, [9](#)  
PBAPL\_GetDocumentInfo, [9](#)  
PBAPL\_GetFieldInfo, [10](#)  
PBAPL\_GetImageBase64, [10](#)  
PBAPL\_GetImageBase64Size, [11](#)  
PBAPL\_GetImageName, [11](#)  
PBAPL\_GetLastError, [11](#)  
PBAPL\_GetSnapshot, [12](#)  
PBAPL\_GetSnapshotBase64, [12](#)  
PBAPL\_GetSnapshotBase64Size, [13](#)  
PBAPL\_GetSnapshotData, [13](#)  
PBAPL\_GetSnapshotSize, [13](#)  
PBAPL\_IsTerminal, [14](#)  
PBAPL\_OpenCaptureDevice, [14](#)  
PBAPL\_RecognizeCurrentFrame, [14](#)  
PBAPL\_SaveImage, [15](#)  
PBAPL\_SaveSnapshot, [15](#)  
PBAPL\_StartAutoCaptureDocument, [16](#)  
PBAPL\_StopAutoCaptureDocument, [16](#)  
PBAPL\_TakeSnapshot, [16](#)