



Ncode SDK

Specification Version 2.0(beta)

NeoLAB Convergence Inc.

Revision History

Ver	Date	Contents
2.00b	04-Oct, 18	New document

Contents

- [1. About this document](#)
- [2. Base concept to Know before getting started](#)
- [3. Preface](#)
- [4. How it works](#)
- [5. Add reference](#)
- [6. Class summary](#)
- [7. How to use Ncode sample application](#)
- [8. Caution](#)
- [9. Generating Ncoded PDF sample with PDF library](#)

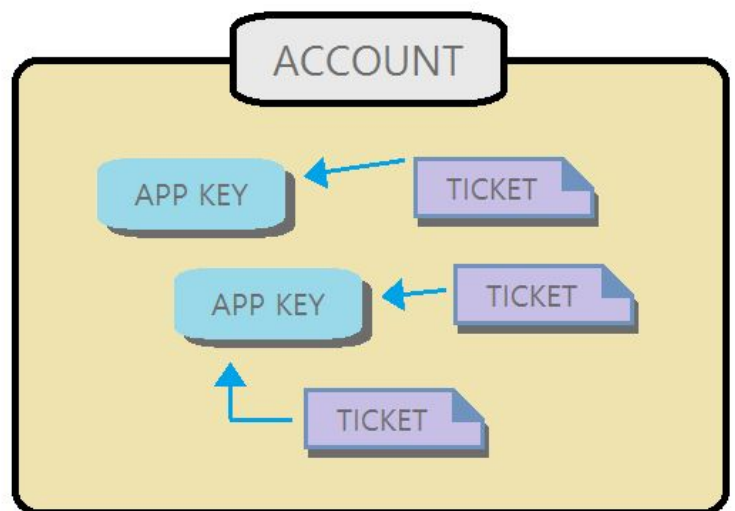
1. About this document

This document is for programming user. This document explains Ncode functions and how to use sample source code.

2. Base concept to Know before getting started

1) Simple process of NeoLAB cloud API service

- 1-1) Create account
- 1-2) Buy tickets
- 1-3) Create app key
- 1-4) Exercise sample app
- 1-5) Test your app key and tickets



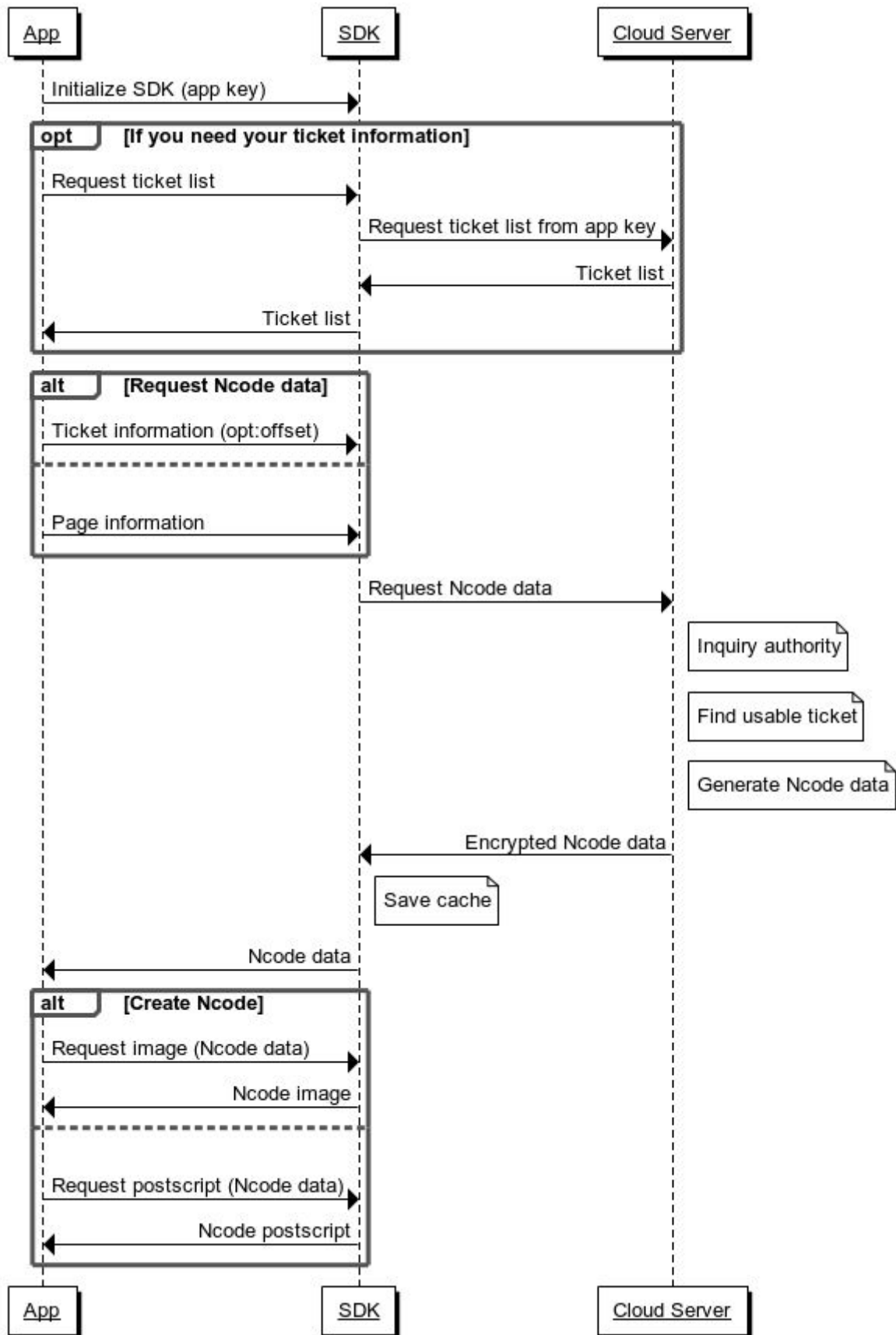
2) Important keywords

- Ncode - NeoLAB convergence short line and dot code
- Account - Account has tickets and app keys
- Ticket - A type of certificates that have a range of purchased Ncode
- App key - Unique identification code for requesting Ncode in application.

3. Preface

Please prepare C# development environment. We recommend Microsoft Visual Studio series.

4. How it works



5. Add reference

NeoLABNcodeSDK.dll
Newtonsoft.Json.dll

6. Class summary

CNcodeSDK class	
This class contains everything about the Ncode SDK's overall processing. Inquiring tickets, Ncode generating and other utility functions.	
bool Init (string appKey, string workingFolder = "", string cacheFolder = "")	Initialize with app key Parameters : appKey - application key workingFolder - the folder in which to store the Ncode image or Ncode postscript file cacheFolder - the folder in which to store the cache Returns : Success > true Failure > false > 370 : Folder setting error
List<TicketInfo> GetTickets ()	Return ticket list from account of app key Returns : Success > TicketInfo list Failure > null > Cannot find app ID Failure > null > 340 : Initialize with app secret key first.
TicketInfo SetStartPageFrom Ticket (TicketInfo ticket, int ownerOffset, int bookOffset, int pageOffset)	Sets the page information to be made based on the selected ticket Parameters : ticket - base ticket ownerOffset - offset owner value from base value bookOffset - offset book value from base value pageOffset - offset page value from base value Returns : Success > Result TicketInfo applied with offset value Failure> null > 350 : Ncode page information is out of your ticket.(owner) Failure> null > 351 : Ncode page information is out of your ticket.(book) Failure> null > 352 : Ncode page information is out of your ticket.(page) Failure> null > 353 : Ncode page information is out of range.

<p>Size GetInchValueFromPaperSize(String paper, bool isLandscape = false)</p>	<p>Get paper inch size(width, height) from paper name like "A4", "Letter"...</p> <p>Parameters : Paper - paper size name isLandscape - if value is true, landscape shape. else portrait</p> <p>Returns : Success > Calculated page width and height inch value Failure > zero size (0, 0) > 360 : Cannot find paper name</p>
<p>int GenerateNcode(out List<NcodePage> pages, TicketInfo startPageInfo, int width, int height, int pageCount)</p>	<p>Request Ncode data.</p> <p>Parameters : pages - NcodePage class list receive data startPageInfo - first page information to create Ncode data width - width of Ncode data to create in inch height - height of Ncode data to create in inch pageCount - the page count to generate</p> <p>Returns : Success > 0 Failure > 340 > Initialize with app secret key first. Failure > 404 > Cannot find app ID Failure > 801 > Generating Ncode failed Failure > 802 > Code type is wrong</p>
<p>NcodeData GenerateNcode(out List<NcodePage> pages, NCODE_TYPE ncodeType, int section, int owner, int book, int page, int width, int height, int pageCount)</p>	<p>Request Ncode data.</p> <p>Parameters : pages - NcodePage class list receive data ncodeType - Ncode type(one of N3C6, G3C6, S1C6, P1C6) section - section number to create owner - owner number to create book - book number to create page - page number to create width - width of Ncode data to create in inch height - height of Ncode data to create in inch pageCount - the page count to generate</p> <p>Returns : Success > 0 Failure > 340 > Initialize with app secret key first. Failure > 404 > Cannot find app ID Failure > 801 > Generating Ncode failed Failure > 802 > Code type is wrong</p>

int GetImage (NcodePage page, int dpi, string filename, bool isBold)	<p>Make Ncode image file from data.</p> <p>Parameters: page - page to make image dpi - dpi of image file(600 or 1200) filename - image file name isBold - bold code option(N3C6)</p> <p>Returns: Success > 0 Failure > 380 > Error occurred while making image.</p>
int GetPostscript (NcodePage page, string filename, double dotScale = 1.0)	<p>Make Ncode postscript file from data. Temporarily support only S1C6 and P1C6.</p> <p>Parameters: page - page to make postscript filename - postscript file name dotScale - dot size(0.8~2.0)</p> <p>Returns: Success > 0 Failure > 341 > DotScale must be in 0.8 ~ 2.0 Failure > 342 > (postscript error stack trace) Failure > 343 > Data type error</p>
string GetLastError ()	<p>Get last error message, if an error occurs.</p> <p>Returns : Error message string</p>
string GetVersion ()	<p>Get current version of Ncode SDK.</p> <p>Returns : Current version of SDK string</p>
class Ticketinfo definition	<pre>public class TicketInfo { public NCODE_TYPE ncodeType; public int section; public int ownerStart; public int ownerSize; public int bookSize; public int bookStart; public int pageStart; public int pageSize; public int period; public string extraInfo; }</pre>

class **NcodePage** definition

```
public class NcodePage
{
    public NCODE_TYPE ncodeType;
    public int section;
    public int owner;
    public int book;
    public int page;
    public double width;
    public double height;
    public string data;
}
```

7. Sample application processing

1) Declare Ncode SDK class.

```
CNcodeSDK sdk = new CNcodeSDK();
```

2) Initialize with app key.

```
sdk.Init("juyhgt54redfv7ujmnhgt5esq0poli");
```

3) Get tickets from pre-set app key.(optional)

```
List<TicketInfo> tickets = sdk.GetTickets();
```

4) Set the start page information.(optional)

```
TicketInfo startPageInfo = sdk.SetStartPageFromTicket(  
    tickets[ticketIndex],  
    ownerOffset,  
    bookOffset,  
    pageOffset);
```

5) Set the page size.(optional)

```
SizeF pageSize = sdk.GetInchValueFromPaperName("A4", false);
```

6-a) Request Ncode data

```
sdk.GenerateNcode(  
    out codeData,  
    startPageInfo,  
    pageSize.Width,      // inch  
    pageSize.Height,     // inch  
    pageCount);
```

6-b) If you know the code range information, you can enter it yourself. Enter code type, section, owner, book and page number instead TicketInfo.

```
sdk.GenerateNcode(  
    out codeData,  
    CNcodeSDK.NCODE_TYPE.N3C6, // Ncode type  
    3,                          // section  
    28,                         // owner  
    10,                         // book  
    1,                          // page  
    6.0,                        // inch  
    8.0,                        // inch  
    pageCount);
```

7-a) Export as image file.

```
sdk.GetImage(codeData[i], 600, outputFilename, false);
```

7-b) Export as postscript file.

```
sdk.GetPosrscript(codeData, outputFilename);
```

8. Caution

The code might not print properly if you directly print the bitmap image that was received with the return value.

In order for the smartpen to correctly recognize Ncode, you should print the pixels as they are without distorting the original. Typically, the printer dithers itself on output to make it easier to see with the naked eye. Because of this, the code is not kept as it is, and the smartpen will not recognize it.

The important thing is that **you have to use the means to print Ncode as it is.**

In addition, even if it can print the original as it is, **the color laser printers that able to handle postscript 2 or higher are recommended.** Because of the case of the inkjet printer smearing.

There are a few things you need to know to get the correct output.

1) Output through Adobe pdf

The Adobe reader(Acrobat reader) has the ability to print the pdf file as it is.

Therefore, if Ncode layer is added to pdf by using pdf library and then printed, Ncode can be output without distortion.

However, when printing, **Ncode must be printed in carbon black color** (corresponding to K in CMYK color of printer), and **background other than Ncode should not contain carbon black.**

This is because the NeoLAB smartpen recognizes only the carbon black color and identifies Ncode. Please take a look at the following parts to help you understand.

2) CMYK and CMYKK '

When printing, four colors of CMYK or five colors of CMYKK 'can be used as the color space.

If you use CMYK, you must use K for Ncode, so if you have a background, you should represent the background color with only CMY color combinations. There is an issue that the background color must be corrected. If you only output Ncode without background, there is no problem.

Therefore, the use of CMYKK '(using non-carbon black) is a good way to get better quality. This is because Ncode can be identified using non-carbon black without modifying the background color.

You need to use a professional printing shop for using CMYKK' color space.

9. Generating Ncoded PDF

1) Ghostscript

You can convert postscript file to Adobe PDF file using "ps2pdf.exe"

If you want to convert on source code, refer below sample code.

```
Process process = new System.Diagnostics.Process();  
process.StartInfo.FileName = "cmd.exe";  
process.StartInfo.Arguments = "/C ps2pdf " + postscriptFilename;  
process.Start();
```

2) Adobe Acrobat pro version can convert postscript to PDF.

3) Datalogics and iTextSharp samples will be updated soon.