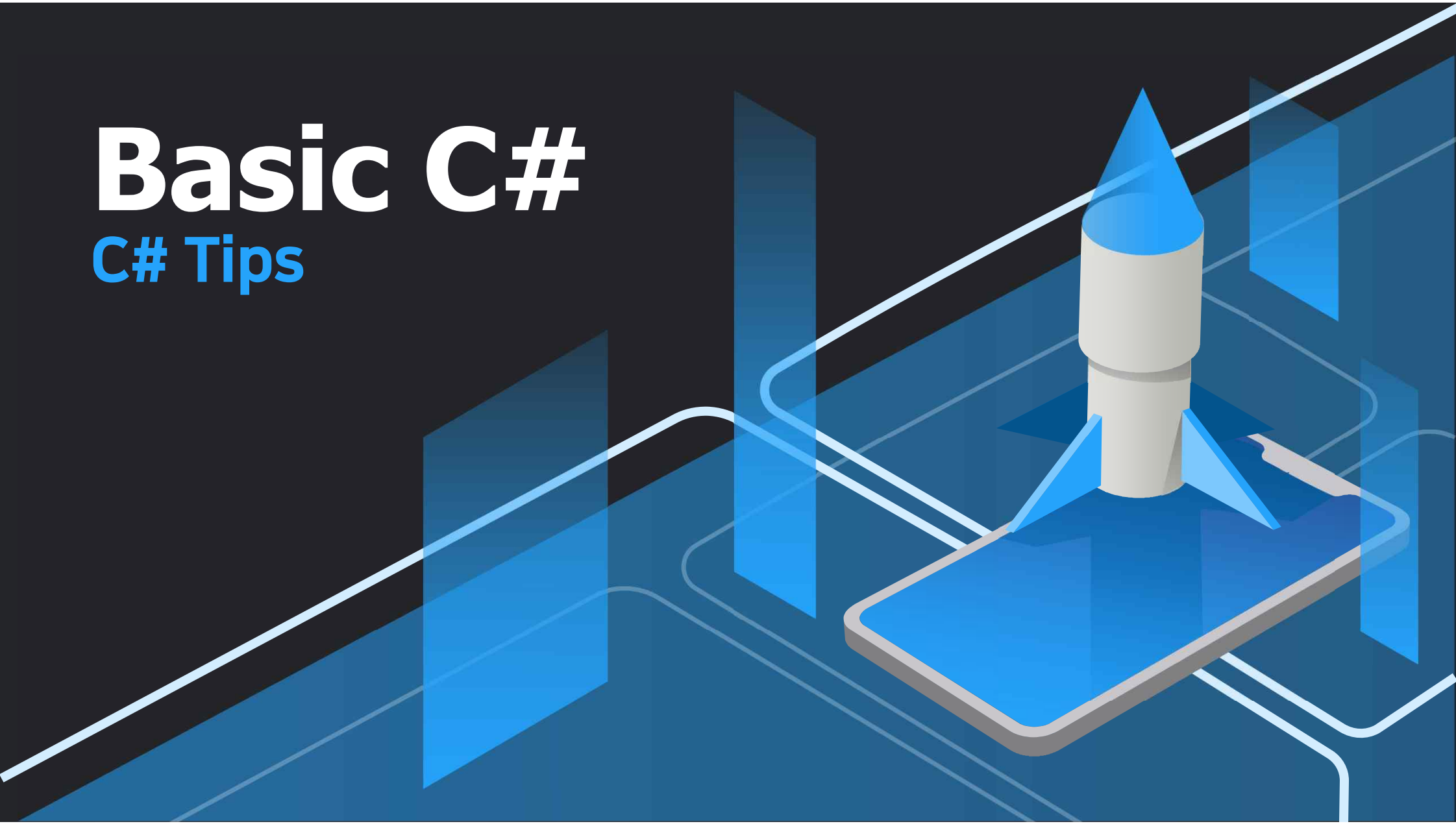


# Basic C#

## C# Tips



---

# Basic C# 목차

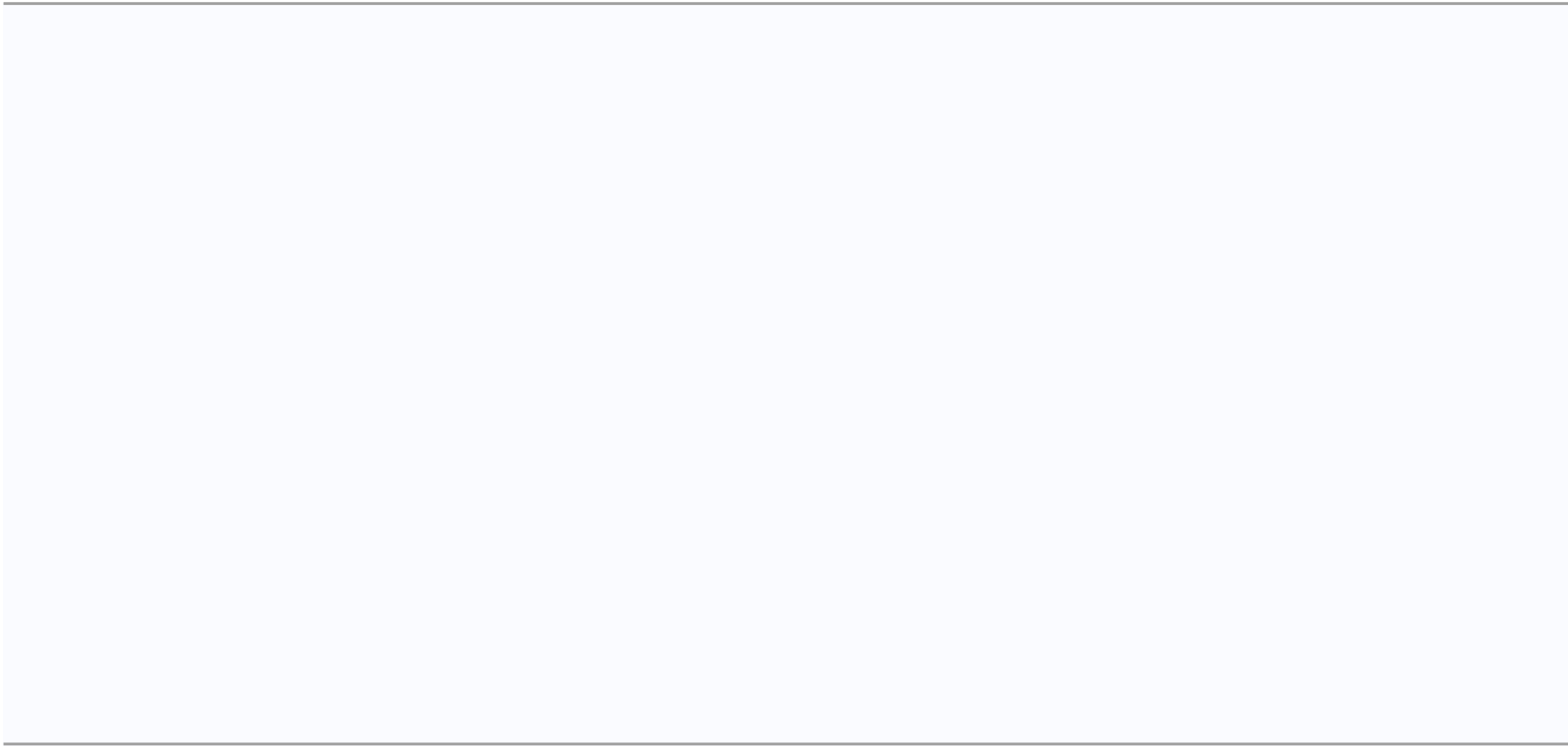
- 동적 컴파일
- .NET Configuration
- Open API 사용

---

# 동적 컴파일

## 1. 컴파일 서비스

A. ComDomProvider



# .NET Configuration

## 1. 프로그램 설정을 담는 파일

### A. ini 파일

```
; 주석입니다
```

```
[database]
```

```
server=1.1.1.1
```

```
port=143
```

```
file="payroll.dat"
```

### vs xml 파일

```
<VirtualBox xmlns="http://www.virtualbox.org/" version="1.16-macosx">
  <Machine uuid="{6df93114-c25d-4365-9055-a95c98dc8e}" name="Hadoop" OSType="U
="Snapshots" lastStateChange="2017-04-28T06:13:50Z">
    <MediaRegistry>
      <HardDisks>
      </HardDisks>
      <DVDImages>
      </DVDImages>
    </MediaRegistry>
    <ExtraData>
      <ExtraDataItem name="GUI/LastCloseAction" value="PowerOff"/>
      <ExtraDataItem name="GUI/LastNormalWindowPosition" value="246,146,800,621"/>
      <ExtraDataItem name="GUI/RestrictedRuntimeDevicesMenuActions" value="HardDrives'
      <ExtraDataItem name="GUI/RestrictedRuntimeMachineMenuActions" value="SaveState,I
      <ExtraDataItem name="GUI/StatusBar/IndicatorOrder" value="HardDisks,OpticalDisks,Fl
haredFolders,Display,VideoCapture,Features,Mouse,Keyboard"/>
    </ExtraData>
  </Machine>
</VirtualBox>
```

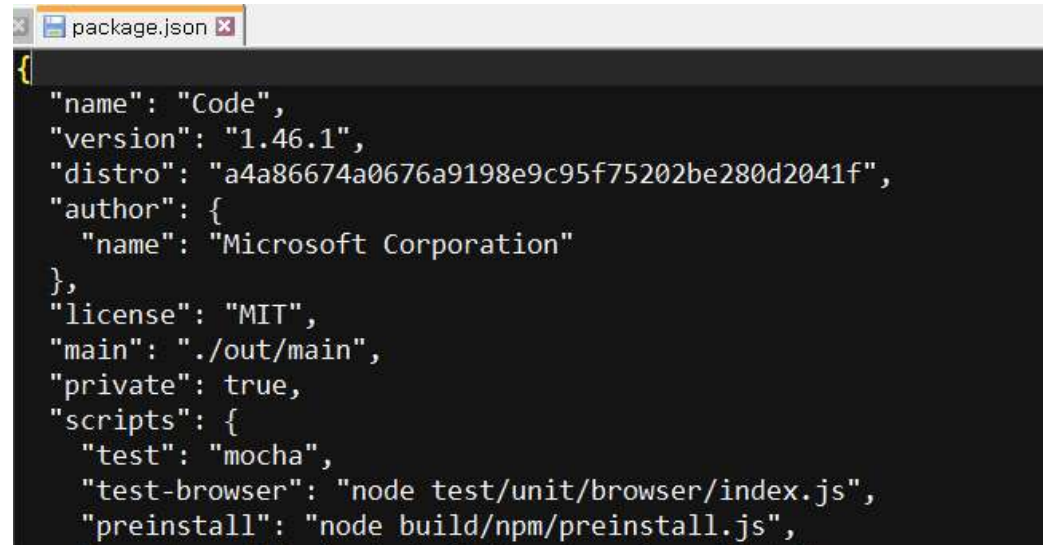
# .NET Configuration

## 1. 프로그램 설정을 담는 파일

- A. json 파일
- B. 윈도우 레지스트리 데이터베이스(파일)

## 2. .NET Framework

- A. .config 파일 사용(XML base : Visual Studio)
- B. .json 파일 사용(Visual Studio Code)

A screenshot of a code editor window showing a file named 'package.json'. The file contains a JSON object with the following properties: 'name' is 'Code', 'version' is '1.46.1', 'distro' is a long alphanumeric string, 'author' is an object with 'name' 'Microsoft Corporation', 'license' is 'MIT', 'main' is './out/main', 'private' is true, and 'scripts' is an object with 'test' 'mocha', 'test-browser' 'node test/unit/browser/index.js', and 'preinstall' 'node build/npm/preinstall.js'.

```
{
  "name": "Code",
  "version": "1.46.1",
  "distro": "a4a86674a0676a9198e9c95f75202be280d2041f",
  "author": {
    "name": "Microsoft Corporation"
  },
  "license": "MIT",
  "main": "./out/main",
  "private": true,
  "scripts": {
    "test": "mocha",
    "test-browser": "node test/unit/browser/index.js",
    "preinstall": "node build/npm/preinstall.js",
  }
}
```

---

# .NET Configuration

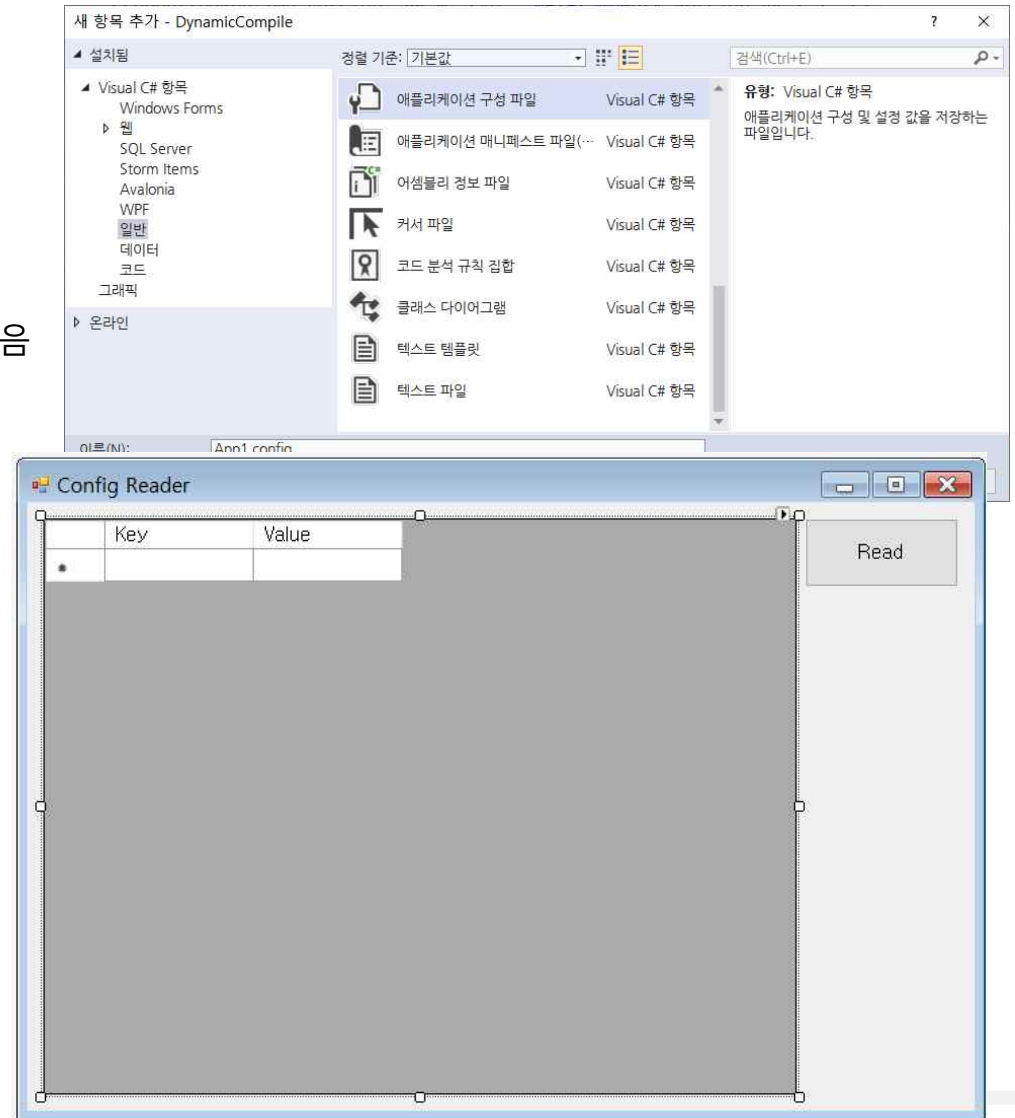
## 1. App.config

- A. 콘솔, 윈폼 or WPF 프로젝트 생성 후 파일 추가(Application Configuration File)
- B. 빌드 후 실행파일에 대한 config 파일 자동 생성
  - a. ex) TestApp.exe 라면 App.config는 TestApp.exe.config 로 변환
- C. ASP.NET 웹 프로그램에서는 Web.config가 사용됨
- D. 프로그램 전반에 영향을 미치는 옵션들을 설정할 때 사용
  - a. 이외 사용자들이 설정하는 사용자 옵션은 별도의 XML이나 DB에 저장, 사용

# .NET Configuration

## 1. 사용

- A. ConfigTestWinApp 원폼 앱 프로젝트 생성 - 자동 추가되어 있음
  - a. 콘솔 앱의 경우 추가필요
- B. 화면 구성
  - a. 버튼, 데이터그리드뷰 추가
  - b. 데이터그리드뷰 열편집
    - 1) Key, Value
- C. 참조 추가
  - a. System.Configuration 추가



# .NET Configuration

## 1. 사용

### A. App.config 수정

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup>
    <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.7.2" />
  </startup>
  <appSettings>
    <add key="SecurityLevel" value="3"/>
    <add key="Locale" value="ko"/>
  </appSettings>
  <connectionStrings>
    <add name="TestDBConnectionString"
          connectionString="Data Source=localhost;Initial Catalog=Test;Integrated Security=True"
          providerName="System.Data.SqlClient"/>
  </connectionStrings>
</configuration>
```



---

# .NET Configuration

## 1. 사용

### A. Read 버튼 클릭이벤트 생성

```
private void button1_Click(object sender, EventArgs e)
{
    List<MyConfig> configs = new List<MyConfig>();

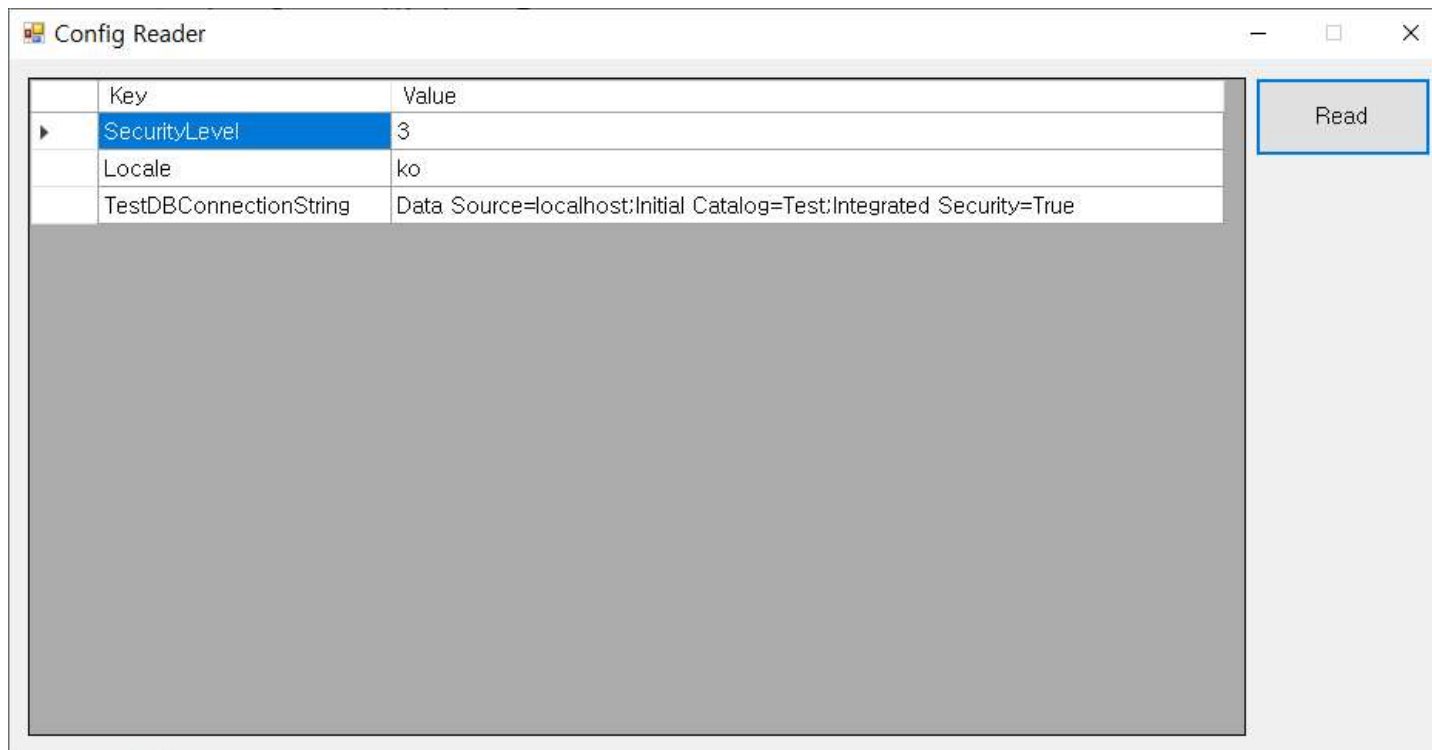
    configs.Add(new MyConfig { Key = "SecurityLevel", Value =
    ConfigurationManager.AppSettings["SecurityLevel"] });
    configs.Add(new MyConfig { Key = "Locale", Value = ConfigurationManager.AppSettings["Locale"] });

    configs.Add(new MyConfig { Key = "TestDBConnectionString", Value =
    ConfigurationManager.ConnectionStrings["TestDBConnectionString"].ConnectionString });
    dataGridView1.DataSource = configs;

    dataGridView1.Columns[0].Width = 190;
    dataGridView1.Columns[1].Width = 540;
}
```

# .NET Configuration

## 1. 실행 결과



---

# OpenAPI 사용

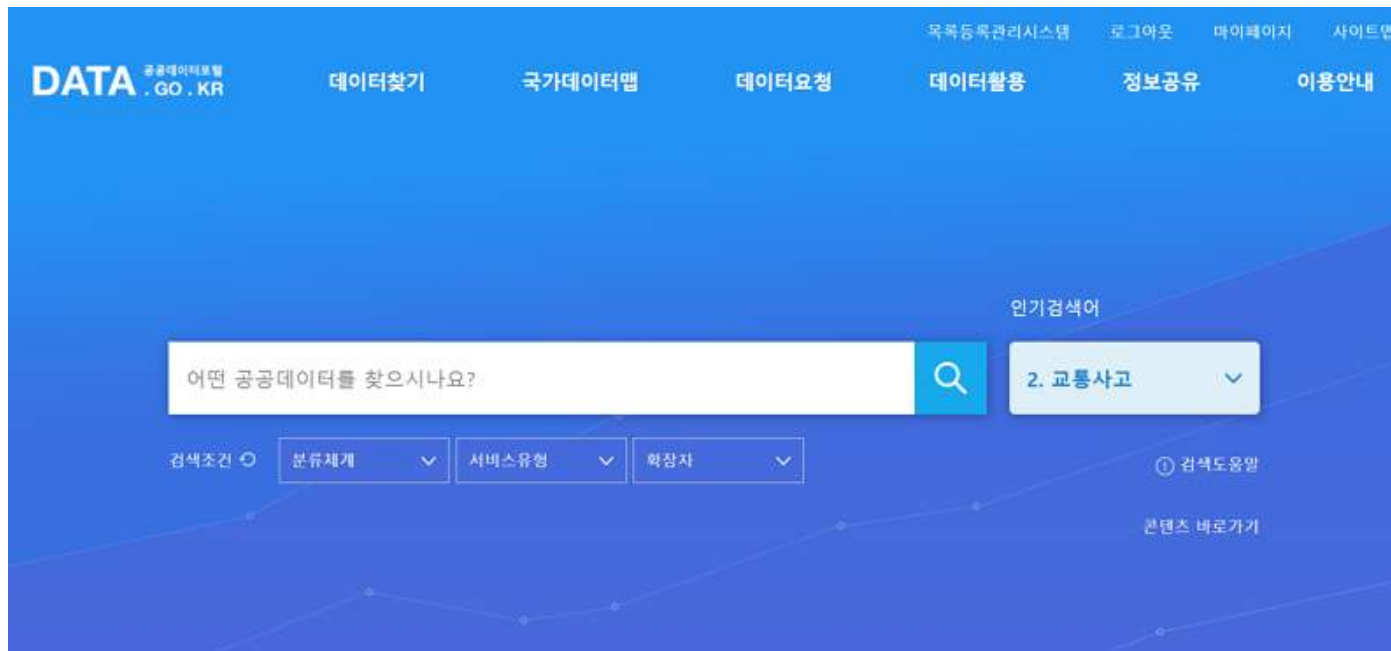
## 1. 개요

- A. 공공데이터포털에서 제공하는 OpenAPI 사용 주식정보 원폼에 디스플레이하기
- B. Visual Studio 2019 Community, MySQL(MariaDB)

# OpenAPI 사용

## 1. 계속

- A. OpenAPI 신청
- B. 공공 데이터 포털 (<https://www.data.go.kr/index.do>)
- C. 회원 가입



# OpenAPI 사용

## 1. 계속

- 로그인 후, 검색 창에 '주식정보서비스' 입력 후 엔터로 검색
- 한국예탁결제원\_주식정보서비스(XML) 확인 후 클릭
- 활용신청 버튼 클릭
- 활용목적 선택
  - 연구(논문 등) 선택 - 학습용
  - 시스템 유형 - 일반 선택
  - 라이선스 표시 - 동의합니다 체크

개발계정

신청 0건 >

신청중인 단계

· 보류 0건

· 반려 0건

활용 2건 >

승인되어 활용중인 단계

· 변경신청 0건

중지0건 >

중지신청하여 운영이 중지된 단계

상세검색 열기 ▾

총2건

재정 금융

한국예탁결제원

활용신청

[승인] 한국예탁결제원\_주식정보서비스

신청일 2020-07-12 만료예정일 2022-07-12

OpenAPI 개발계정 신청

XML 한국예탁결제원\_주식정보서비스

제공기관	한국예탁결제원	서비스유형	REST
심의여부	자동승인	신청유형	개발계정   활용신청
처리상태	신청	활용기간	승인일로부터 24개월 간 활용가능

공공데이터 제공제도

\*공공데이터를 위치정보를 포함한 서비스를 사용하고자 하는 사업자는 '위치정보의 보호 및 이용 등에 관한 법률'에 따라 방송통신위원회에 '위치정보서비스 허가'를 받거나 '위치기반 서비스사업 신고'를 하여야 합니다.

\*이에 해당하는 사업자인 경우에는 첨부파일에 '위치기반서비스사업신고필증'을 첨부해 주시기 바랍니다.

\*활용신청 시 '위치기반서비스사업신고필증'이 등록되지 않으면 반려가 될 수 있으니 참고하시기 바랍니다.

활용목적 선택 

\*표시는 필수 입력항목입니다.

\*활용목적

· 웹 사이트 개발

· 앱개발 (모바일 솔루션 등)

· 기타

· 참고자료

· 연구(논문 등)

0/250

첨부파일

파일 선택

Drag & Drop으로 파일을 선택 가능합니다.

시스템유형 선택

시스템 유형

· 일반

· 서버구축

[안내] 서버 구축 여부에 따라 시스템 유형을 입력하실 수 있습니다.

\*일반 : OPEN API 서비스를 호출하여 응답 받은 결과값을 서버에 저장하지 않고 사용할 경우 (서버 미구축)

\*서버구축 : OPEN API 서비스를 호출하여 응답 받은 결과값을 서버에 저장하거나 DB화하여 사용할 경우

# OpenAPI 사용

## 1. 계속

### A. 승인여부 확인 및 인증키 복사

#### 서비스정보

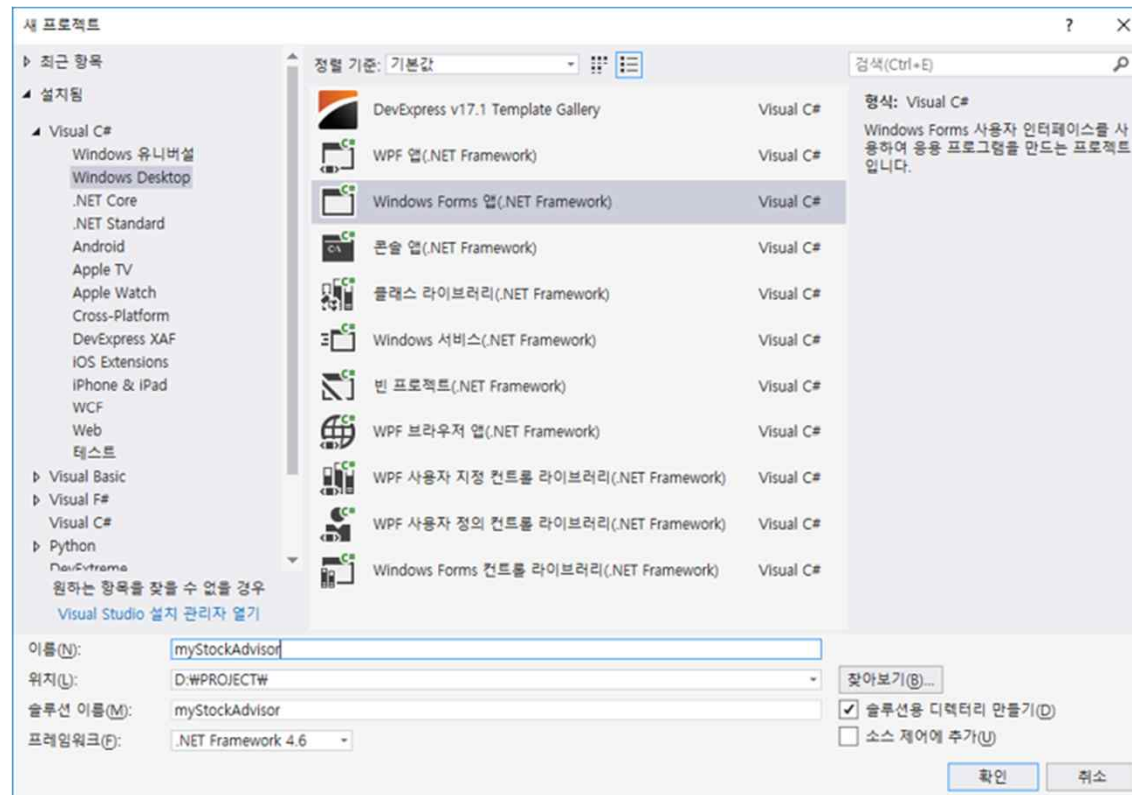
일반 인증키 (UTF-8)	hPw%3D%3D
End Point	http://api.seibro.or.kr/openapi/service/StockSvc?_wadl&type=xml
데이터포맷	XML

### B. '종목명칭으로 주식종목으로 및 기본정보 조회' 사용

# OpenAPI 사용

## 1. 시작

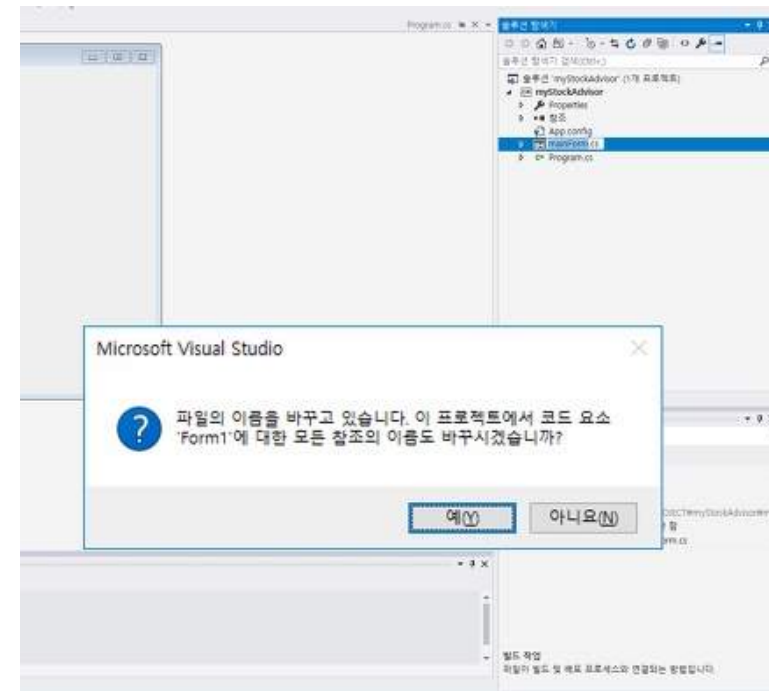
### A. MyStockSystem 원품 프로젝트 생성



# OpenAPI 사용

## 1. 계속

A. Form1.cs 파일명 MainForm.cs로 변경





# OpenAPI 사용

## 1. 계속

### A. MainForm 화면 디자인

#### a. 속성 변경

- 1) Text → 주식분석 시스템
- 2) Size → (1280, 720) 적당히~

### B. NuGet 패키지

#### a. MetroFramework 추가

### C. 화면 구성

#### a. MetroTitle 3개



# OpenAPI 사용

## 1. 계속

### A. SearchItemForm 화면 구성

- a. MetroTab
- b. SplitContainer
- c. TextBox
- d. Button
- e. DataGridView

The screenshot displays a Windows application window titled "종목 검색" (Stock Search). It features a Metro-style interface with a blue title bar. Below the title bar, there are two tabs: "주식기초정보수집" (Stock Basic Information Collection) and "해당정보수집" (Collect corresponding information). The "주식기초정보수집" tab is active. The main content area is divided into two sections. The top section contains a search bar with a "Search" button. The bottom section is a large, empty gray rectangle, likely representing a DataGridView. A small blue button with a double-left arrow is located in the bottom right corner of the window.

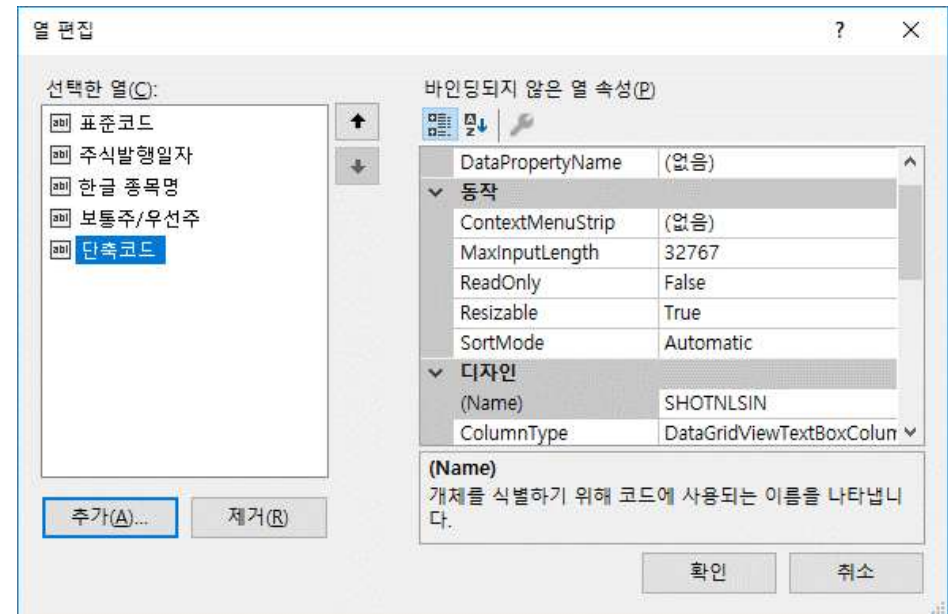
# OpenAPI 사용

## 1. 계속

### A. SearchItemForm 화면 구성

#### a. DataGridView 열 추가

- 1) 이름 ISIN, 머리글 텍스트 표준코드
- 2) ISSUDT, 주식발행일자
- 3) KORSECNAME, 한글종목명
- 4) SECNKACDNM, 보통주/우선주
- 5) SHOTNISIN, 단축코드



# OpenAPI 사용

## 1. 계속

### A. SearchItemForm 클래스 - 검색버튼 클릭 이벤트 핸들러

```
WebClient wc = null;
XmlDocument doc = null;

wc = new WebClient() { Encoding = Encoding.UTF8 };
doc = new XmlDocument();

StringBuilder str = new StringBuilder();
str.Append("http://api.seibro.or.kr/openapi/service/StockSvc/getStkIsinByNmN1");
str.Append("?serviceKey=qzyS5rrq0D4aHe%2B8lcYGp0LCf4hA0XrMG%2BbCDrg2TJMrBhTu82nhMSw71WQsM4CUj1EJc%2B3%2Fo%2BT9K0EZFJihPw%3D%3D"); //인증키
str.Append("&secnNm=" + TxtSearchItem.Text); //종목명
str.Append("&pageNo=1"); //페이지 수
str.Append("&numOfRows=200"); //읽어올 데이터 수
str.Append("&martTpcd=11"); //주식시장종류 : 11은 유가증권시장

string xml = wc.DownloadString(str.ToString());
doc.LoadXml(xml);

XmlElement root = doc.DocumentElement;
XmlNodeList items = doc.GetElementsByTagName("item");
```

---

# OpenAPI 사용

## 1. 계속

### A. SearchItemForm 클래스 - 검색버튼 클릭 이벤트 핸들러

```
DgvStocks.Rows.Clear();

foreach (XmlNode item in items)
{
    DgvStocks.Rows.Add(item["isin"].InnerText, item["issuDt"].InnerText,
        item["korSecnNm"].InnerText, item["secnKacdNm"].InnerText, item["shotnIsin"].InnerText);
}

DgvStocks.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.DisplayedCells;
```

# OpenAPI 사용

## 1. 계속

### A. 결과 실행

종목 검색

주식기초정보수집 해당정보수집

삼성 Search

	표준코드	주식발행일자	한국종목명	보통주/우선주	단축코드
▶	KR7314250002	20190108	삼성KODEX 미국FANG플러스증권상장지수투자신탁(주식-파생형)(H)	보통주	314250
	KR7244620001	20160512	삼성KODEX모멘텀PLUS증권상장지수투자신탁[주식]	보통주	244620
	KR7321410003	20190603	삼성KODEX 모닝스타멀티에셋하이인컴증권상장지수투자신탁 [채권혼합-파생재간접형](H)	보통주	321410
	KR7114260003	20090728	삼성 KODEX 국고채3년 증권상장지수투자신탁[채권]	보통주	114260
	KR7114800006	20090914	삼성KODEX인버스증권상장지수투자신탁	보통주	114800
	KR7280940008	20171107	삼성KODEX골드선물인버스특별자산상장지수투자신탁[금-파생형](H)	보통주	280940
	KR7275280006	20170710	삼성KODEX MSCI모멘텀증권상장지수투자신탁[주식]	보통주	275280
	KR7291890002	20180321	삼성KODEX SGX MSCI EM선물증권상장지수투자신탁 [주식-파생형](H)	보통주	291890
	KR7292190006	20180323	삼성 KODEX KRX300증권상장지수투자신탁[주식형]	보통주	292190
	KR7251340006	20160809	삼성KODEX코스닥150인버스증권상장지수투자신탁[주식-파생형]	보통주	251340
	KR7138920004	20110311	삼성KODEX공선물특별자산상장지수투자신탁[공-파생형](H)	보통주	138920
	KR7226980001	20150827	삼성KODEX200중소형증권상장지수투자신탁[주식]	보통주	226980
	KR7010141000	19941028	삼성중공업1우	우선주	010145
	KR7005930003	19750611	삼성전자	보통주	005930
	KR7000810002	19750630	삼성화재해상보험	보통주	000810
	KR7337120000	20191113	삼성 KODEX Fn 멀티팩터증권상장지수투자신탁[주식]	보통주	337120
	KR7337140008	20191113	삼성KODEX 코스피대형주증권상장지수투자신탁[주식]	보통주	337140
	KR7337150007	20191113	삼성KODEX 200exTOP 증권상장지수투자신탁[주식]	보통주	337150
	KR7337160006	20191113	삼성KODEX 200ESG 증권상장지수투자신탁[주식]	보통주	337160
	KR7010261006	19990310	삼성메디슨1우	우선주	010265

---

# 수고 많으셨습니다

추가 문의사항이 있을 경우, 메일로 여쭙주시기 바랍니다.

[personar95@naver.com](mailto:personar95@naver.com)

---

