



Let's Gorail



장승민, 김동규



GORAIL

목차



개요

Language



IDE



API



Fonts & colors used

나눔 고딕
나눔 바른 고딕

Fonts color
Black & white

BG

#ECF1F4

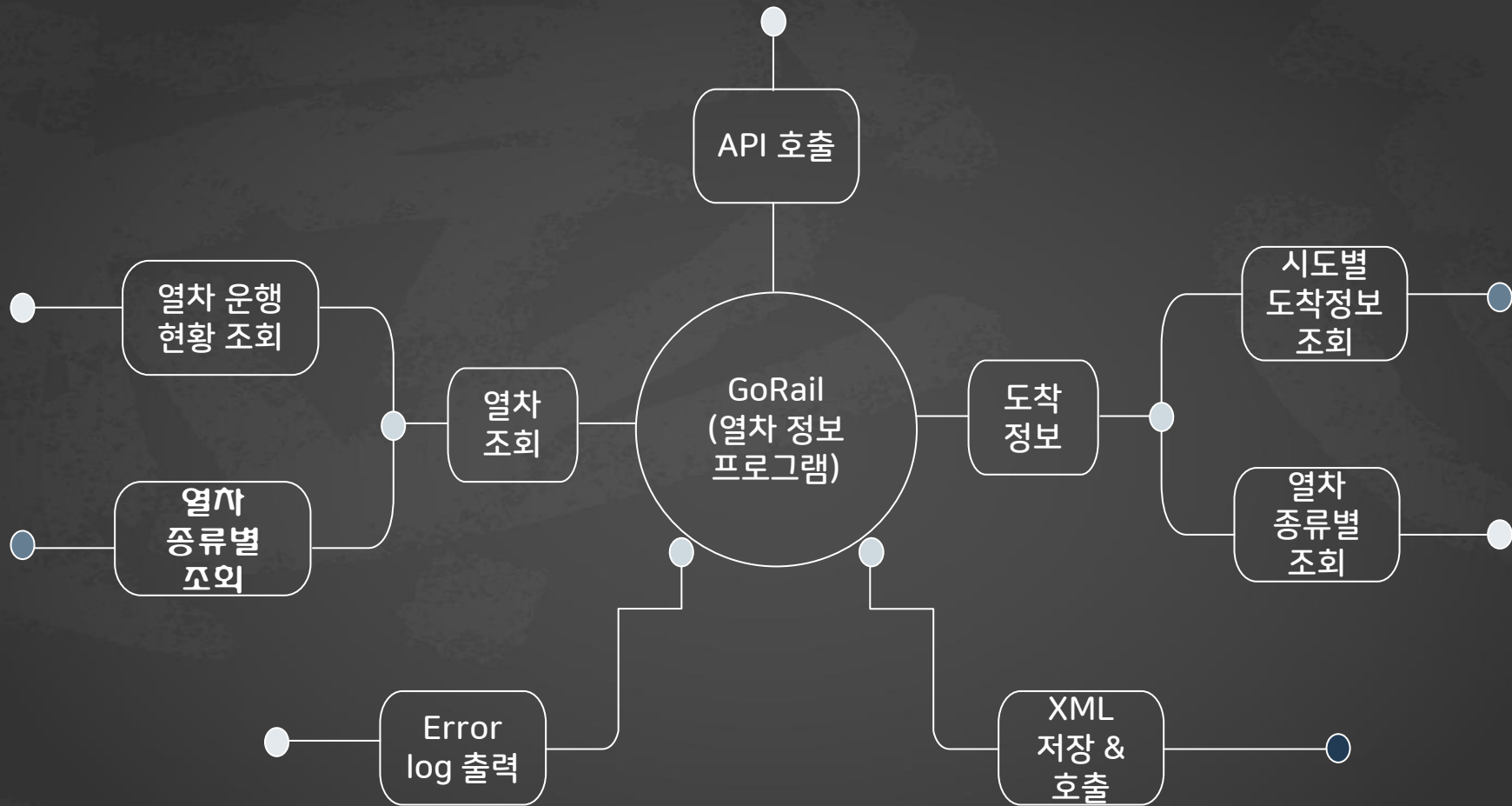
Button

#2E74D4

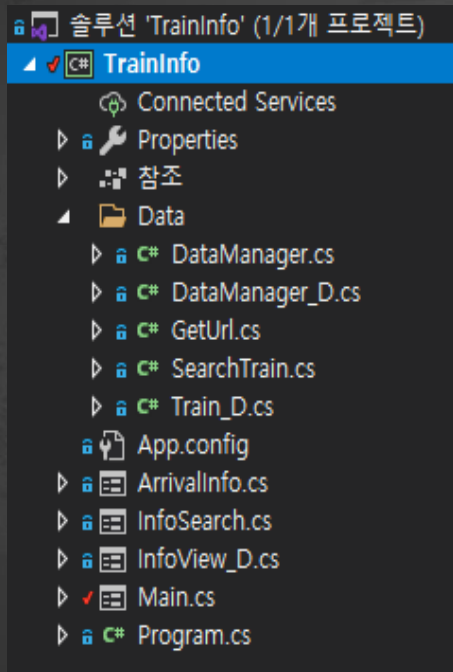
Hover

#96D2FF





프로젝트 목록



Form

Main.cs - HOME 화면, 열차 조회 및 도착정보 버튼

InfoSearch.cs - 열차 조회

InfoView - 도착 정보 조회

ArrivalInfo.cs - 도착정보 조회 전, 광역시 선택창

Class

DataManager_D.cs - 도착 정보에 사용하는 API 호출 및 로그 저장, XML 저장 호출 등의 메서드 포함

DataManager.cs - 로그 저장 등의 메서드 포함

GetUrl.cs - 열차 조회에 사용 하는 API 호출 메서드 포함

SearChTrain.cs - 열차 조회의 데이터를 저장할 객체 생성

Train_D.cs-도착 정보의 데이터를 저장할 객체 생성



상세 기능

메인 화면

코레일 열차
영상 출력



도착 정보
버튼

열차 조회
버튼
Mouse Hover 시 색상 변경





영상 출력

```
VideoCapture video;
Mat frame = new Mat();
//조 1개
private void Main_Load(object sender, EventArgs e)
{
    //영상 출력 opencvsharp
    try
    {
        video = new VideoCapture(@"home3.mp4");
    }
    catch
    {
        timer1.Enabled = false;
    }
}

//기차영상 tick 출력
//조 1개
private void timer1_Tick(object sender, EventArgs e)
{
    video.Read(frame);
    if (video.PosFrames == video.FrameCount) //마지막 프레임일때
        video.Open(@"home3.mp4");//다시 실행
    //속도 조절
    if (video.PosFrames < (video.FrameCount / 2)*1.2) {
        if(timer1.Interval>10)
            timer1.Interval -= 1;
    }
}
```

영상출력을 위해 opencvsharp 사용
Timer에서 영상 진행 속도 조절



영상의 실제 속도로 출력 할수 있는
방안 검토, 소리 출력 또한 될 수
있도록 수정 필요

열차 조회

⇅버튼 클릭 시 출발역, 도착역의 텍스트가 바뀜

조회 버튼 클릭 시 전체 열차 정보가 dataGridView에 출력 됨

조회할 날짜를
클릭하면
위의 출발일
텍스트가 해당
날짜로 변경됨

역 이름 입력 시
자동완성 기능이
적용됨

현재 시각을
나타냄

열차 조회

출발일 2021.04.30

출발역 서울

도착역 동대구

조회

☒ 전체
☐ KTX / SRT
☐ ITX - 청춘
☐ 새마을호 / ITX-새마을
☐ 무궁화호 / 누리로
☐ 통근열차

남은 시간	출발 역	출발 시간	도착 시간	기차 종류	기차 번호	소요 시간
4 분	서울	12:41	14:20	KTX	243	1 시간 39 분
13 분	서울	12:50	14:36	KTX	211	1 시간 46 분
23 분	서울	13:00	14:55	KTX-산천	241	1 시간 55 분
23 분	서울	13:00	14:55	KTX-산천	31	1 시간 55 분
29 분	서울	13:06	17:10	무궁화호	1213	4 시간 4 분
33 분	서울	13:10	15:02	KTX	103	1 시간 52 분
43 분	서울	13:20	15:10	KTX	33	1 시간 50 분
1 시간 3 분	서울	13:40	15:20	KTX	213	1 시간 40 분
1 시간 15 분	서울	13:52	15:37	KTX	83	1 시간 45 분
1 시간 19 분	서울	13:56	15:46	KTX	35	1 시간 50 분
1 시간 22 분	서울	13:59	17:26	ITX-새마을	1009	3 시간 27 분
1 시간 29 분	서울	14:06	18:02	무궁화호	1215	3 시간 56 분

현재 시각 : 2021-04-30 12:38:01

각 버튼 클릭 시 해당
열차들에 관한 열차
정보만
dataGridView에
출력됨

API 정보를 기반으로
남은 시간 및
소요시간을 계산해
출력



열차 정보 조회 API 호출

```
public void getStrtpntAlocFndTrainInfo(string date, string depPlaceId, string arrPlaceId, string[] trainName)
{
    string url = "http://openapi.tago.go.kr/openapi/service/TrainInfoService/getStrtpntAlocFndTrainInfo"; // URL
    url += "?ServiceKey=gsClkYQY5L7abfYL7oyicjcCbDBXKny%2BiAx8pSuSBvevP%2B9XASascNCaoWJz%2F89mmb%2BHM53e2xib0743Dr%2BVVQ%3D%3D";
    url += "&numOfRows=500";
    url += "&pageNo=1";
    url += "&depPlaceId=" + depPlaceId;
    url += "&arrPlaceId=" + arrPlaceId;
    url += "&depPlandTime=" + date;

    st.Clear();
    foreach (var item in request(url).Descendants("item")){...
```



호출된 API정보를 XML 형식으로 전환

```
public XElement request(string url)
{
    var request = (HttpWebRequest)WebRequest.Create(url);
    request.Method = "GET";

    string results = string.Empty;
    HttpWebResponse response;
    using (response = request.GetResponse() as HttpWebResponse)
    {
        StreamReader reader = new StreamReader(response.GetResponseStream());
        results = reader.ReadToEnd();
    }
    XElement resultsXElement = XElement.Parse(results);
    return resultsXElement;
}
```



도시 코드 정보 호출

```
public Dictionary<string, string> getCtyCodeList()
{
    string url = "http://openapi.tago.go.kr/openapi/service/TrainInfoService/getCtyCodeList"; // URL
    url += "?ServiceKey=gsClkYQY5L7abfYL7oyicjCcbDBXKny%2BiAx8pSuSBvevP%2B9XASascNCaoWJz%2F89mmmb%2BHM53e2xib0743Dr%2BVVQ%3D%3D";

    Dictionary<string, string> city = new Dictionary<string, string>();

    foreach (var item in request(url).Descendants("item"))
    {
        city.Add(item.Element("citycode").Value, item.Element("cityname").Value);
    }

    return city;
}
```



열차 종류 정보 호출

```
public Dictionary<string, string> getVhcleKndList()
{
    string url = "http://openapi.tago.go.kr/openapi/service/TrainInfoService/getVhcleKndList"; // URL
    url += "?ServiceKey=gsClkYQY5L7abfYL7oyicjCcbDBXKny%2BiAx8pSuSBvevP%2B9XASascNCaoWJz%2F89mmmb%2BHM53e2xib0743Dr%2BVVQ%3D%3D";

    Console.WriteLine(request(url));
    Dictionary<string, string> vehicle = new Dictionary<string, string>();
    foreach (var item in request(url).Descendants("item"))
    {
        vehicle.Add(item.Element("vehiclekndid").Value, item.Element("vehiclekndnm").Value);
    }
    return vehicle;
}
```



기사역 코드 정보 호출

```
public Dictionary<string, string> getCityAcctoTrainSttnList(string cityCode)
{
    string url = "http://openapi.tago.go.kr/openapi/service/TrainInfoService/getCityAcctoTrainSttnList"; // URL
    url += "?ServiceKey=gsClkYQY5L7abfYL7oyicjcCbDBXKny%2BiAx8pSuSBvevP%2B9XASascNCaoWJz%2F89mmB%2BHM53e2xib0743Dr%2BVVQ%3D%3D";
    url += "&numOfRows=200";
    url += "&pageNo=1";
    url += "&cityCode=" + cityCode;

    Dictionary<string, string> node = new Dictionary<string, string>();

    foreach (var item in request(url).Descendants("item"))
    {
        node.Add(item.Element("nodeid").Value, item.Element("nodename").Value);
    }

    return node;
}
```

도시 코드를
매개변수로 해당
도시 코드에 있는
역만 호출할 수
있어 API 호출
빈도가 많아지는
단점이 있음.



모든 기사역 정보 호출

```
public Dictionary<string, string> getAllTrain()
{
    String allTrain = File.ReadAllText(@"./AllTrainCode.xml");
    XElement carXElement = XElement.Parse(allTrain);

    Dictionary<string, string> allTrains = new Dictionary<string, string>();

    foreach (var item in carXElement.Descendants("item"))
    {
        allTrains.Add(item.Element("nodeid").Value, item.Element("nodename").Value);
    }

    return allTrains;
}
```

위의 단점을 보완하기 위해 역
정보는 고정된 값이라 판단하여
모든 도시에 있는 역 정보를
XML파일로 생성하여 호출하여
사용



도착정보

기차역(출발)	서울	조 회	
기차역(도착)	대구		
남은시간	대구	발지	출발
도착	KTIX	서울	04월30
2분	ITX-새마을	서울	04월30

도착 지역 선택 후 그
지역 도착역만
ComboBox로 출력

마우스 HOVER 기능
적용

지역선택

전체	서울	인천	대전
대구	부산	울산	광주

도착정보

각 버튼 클릭 후 조회시 해당 열차들에
관한 열차 정보만 dataGridView에
출력됨

선택한 지역 도착역만
ComboBox로 출력,
단어자동 완성기능

API 정보로 남은
시간 계산 후 출력
현 시각과
도착시간이 같다면
남은 시간은
"도착"으로 출력

도착정보

기차역(출발) 서울

기차역(도착) 동대구

조회

☒ 전체
 ☐ ITX-청춘
 ☐ KTX/SRT
 ☐ 새마을/ITX-새마을
 ☐ 무궁화/누리로
 ☐ 통근열차

동대구역 도착정보

2021-04-30 12:40:01

남은시간	도착	출발	출발 시간	도착지	도착 시간	기차 역명	열차 번호
도착	KTX	서울	04월30일 10:14	동대구	04월30일 12:40	동대구역	123
2분	ITX-새마을	서울	04월30일 09:15	동대구	04월30일 12:42	동대구역	1003
6분	KTX	서울	04월30일 11:00	동대구	04월30일 12:46	동대구역	25
66분	KTX	서울	04월30일 12:00	동대구	04월30일 13:46	동대구역	27
85분	무궁화호	서울	04월30일 09:56	동대구	04월30일 14:05	동대구역	1209
93분	KTX	서울	04월30일 12:27	동대구	04월30일 14:13	동대구역	29
100분	KTX	서울	04월30일 12:41	동대구	04월30일 14:20	동대구역	243
103분	무궁화호	서울	04월30일 10:26	동대구	04월30일 14:23	동대구역	1301
108분	ITX-새마을	서울	04월30일 11:04	동대구	04월30일 14:28	동대구역	1005
116분	KTX	서울	04월30일 12:50	동대구	04월30일 14:36	동대구역	211
135분	KTX-산천	서울	04월30일 13:00	동대구	04월30일 14:55	동대구역	241
135분	KTX-산천	서울	04월30일 13:00	동대구	04월30일 14:55	동대구역	31
142분	KTX	서울	04월30일 13:10	동대구	04월30일 15:02	동대구역	103
150분	KTX	서울	04월30일 13:20	동대구	04월30일 15:10	동대구역	33
156분	ITX-새마을	서울	04월30일 11:43	동대구	04월30일 15:16	동대구역	1007
160분	KTX	서울	04월30일 13:40	동대구	04월30일 15:20	동대구역	213
168분	무궁화호	서울	04월30일 11:15	동대구	04월30일 15:28	동대구역	1211
177분	KTX	서울	04월30일 13:52	동대구	04월30일 15:37	동대구역	83

도착역 조회시 이름
출력 & 현재 시간
timer로 출력

dataGridView에
데이터소스 연결 후
데이터 출력



열차 정보 조회 API 호출 및 XML 저장

```
//지역 코드 정보
참조 2개
public static void getCtyCodeList()
{
    results = "";

    try
    {
        results = File.ReadAllText(@"./dg_xml/ctyCode.xml");
        Console.WriteLine("ctyCode.xml load 성공");

        XElement ctyCode = XElement.Parse(results);
        //도시 코드 저장
        foreach (var item in ctyCode.Descendants("item")){...}
        //확인용
        foreach (var item in trainGradeCode){...}
    }
    catch (Exception ex)
    {
        string url = "http://openapi.tago.go.kr/openapi/service/TrainInfoService/getCtyCodeList"; // URL
        url += "?ServiceKey=" + "01L6R0F8vitdpbkttDCEx3Uxse07CQ1RRK3plz2BdAkBSYkIESNMfTtYmQkz2BPUDXLPQfvB3iGXJYvPOS2brP4gQX3Dx3D";

        request(url); //api 호출

        XElement ctyCode = XElement.Parse(results);
        File.WriteAllText(@"./dg_xml/ctyCode.xml", ctyCode.ToString());

        writeLog(ex.Message);
        writeLog(ex.StackTrace);
        getCtyCodeList(); //다시 호출
    }
}
```

하루 트래픽이 제한되어 있어
API 정보 중 변하지 않는 고정된
값인 지역, 기차역 코드 정보 등을
처음에만 URL 호출 하고
그후에는 XML로 저장한 데이터
호출

XML파일이 없을 시 AIP다시
호출하여 XML저장을 하는데
인터넷 연결 확인을 할 수 있게
보완할 필요 있음



도착 역, 열차 종류 별 도착정보 출력

```
//기차 종류 선택 코드 정보 입력
참조 1개
private List<string> selectTrain()
{
    List<string> st = new List<string>();
    st.Clear();

    if (radio1.Checked)
    {
        st.Add("ALL");
    }

    else if (radio2.Checked)
    {
        st.Add(DataManager.D.ITX_CHUNGCHUN1);
        st.Add(DataManager.D.ITX_CHUNGCHUN2);
    }

    else if (radio3.Checked)
    {
        st.Add(DataManager.D.KTX);
        st.Add(DataManager.D.SRT);
        st.Add(DataManager.D.KTX_LSEEM);
    }

    else if (radio4.Checked)
    {
        st.Add(DataManager.D.SAEMAEUL);
        st.Add(DataManager.D.ITX_SAEMAEUL);
    }

    else if (radio5.Checked)
    {
        st.Add(DataManager.D.MUGUNHWA);
        st.Add(DataManager.D.NURIRO);
    }

    else if (radio6.Checked)
    {
        st.Add(DataManager.D.COMMUTER);
    }

    return st;
}
```

- ☐ 전체
- ☐ ITX-청춘
- ☒ KTX/SRT
- ☐ 새마을/ITX-새마을
- ☐ 무궁화/누리로
- ☐ 통근열차

기차역(출발)	<input type="text" value="서울"/>	<input type="button" value="조회"/>
기차역(도착)	<input type="text" value="동대구"/>	

남은시간	차량 종류	출발지	출발
11분	KTX	서울	05월
31분	KTX	서울	05월
55분	KTX	서울	05월
90분	KTX	서울	05월
108분	KTX	서울	05월
135분	KTX	서울	05월
157분	KTX	서울	05월
177분	KTX	서울	05월
181분	KTX	서울	05월
201분	KTX	서울	05월
231분	KTX	서울	05월
241분	KTX	서울	05월
245분	KTX	서울	05월
260분	KTX	서울	05월
283분	KTX	서울	05월
296분	KTX	서울	05월
326분	KTX	서울	05월
330분	KTX	서울	05월

고정된 차량 코드정보를 이용하여
원하는 차량정보를 필터 할 수
있도록 기능 추가

API 정보 및 제한된 트래픽의
한계로 출발역을 선택해야 한다는
점 보완 필요





열차 도착정보 저장

```
//열차정보 저장
foreach (var item in ShowInfo.Descendants("item"))
{
    string tmpTraingradename = item.Element("traingradename").Value; //차량종류명
    DateTime tmpDepplandtime = DateTime.ParseExact(item.Element("depplandtime").Value, "yyyyMMddHHmmss", null); //차량출발시간
    DateTime tmpArrplandtime = DateTime.ParseExact(item.Element("arrplandtime").Value, "yyyyMMddHHmmss", null); //도착시간
    string tmpDepplacename = item.Element("depplacename").Value; //출발지
    string tmpArrplacename = item.Element("arrplacename").Value; //도착지
    string tmpTrainno = item.Element("trainno").Value; //열차번호
    string tmpNodeName = item.Element("arrplacename").Value + "역"; //도착지
    //도착시간 구하는
    string test = DateTime.Now.ToString("yyyyMMddHHmmss");
    DateTime now = DateTime.ParseExact(test, "yyyyMMddHHmmss", System.Globalization.CultureInfo.InvariantCulture);
    TimeSpan timeSpan = tmpArrplandtime - now;
    int timeHoure = timeSpan.Hours;
    int timeMinutes = timeSpan.Minutes;
    Console.WriteLine("시간 : " + item.Element("arrplandtime"));

    Train_D td = new Train_D()
    {
        traingradename = tmpTraingradename,
        depplandtime = item.Element("depplandtime").Value.Substring(4, 2) + "월 " +
            item.Element("depplandtime").Value.Substring(6, 2) + "일 " +
            item.Element("depplandtime").Value.Substring(8, 2) + ":" +
            item.Element("depplandtime").Value.Substring(10, 2),
        arrplandtime = item.Element("arrplandtime").Value.Substring(4, 2) + "월 " +
            item.Element("arrplandtime").Value.Substring(6, 2) + "일 " +
            item.Element("arrplandtime").Value.Substring(8, 2) + ":" +
            item.Element("arrplandtime").Value.Substring(10, 2),
        depplacename = tmpDepplacename,
        arrplacename = tmpArrplacename,
        trainno = tmpTrainno,
        nodeName = tmpNodeName,
        outTime = (timeHoure * 60) + timeMinutes + "분",
        timeCheck = (timeHoure * 60) + timeMinutes
    };
    if ((timeHoure * 60) + timeMinutes >= 0)
    {
        if ((timeHoure * 60) + timeMinutes == 0)
        {
            td.outTime = "도착";
        }
    }
}
```

API 데이터 저장 및 도착 남은 시간을
구해서 리스트에 저장
String 형식을 DateTime형식으로
변형하여 TimeSpan 활용

조회 했던 API는 XML로 저장

- [20210428]ShowInfo[용산][서울].xml
- [20210428]ShowInfo[화명][서울].xml
- [20210430]ShowInfo[부산][서울].xml
- [20210430]ShowInfo[서울][대구].xml
- [20210430]ShowInfo[서울][동대구].xml
- [20210503]ShowInfo[서울][동대구].xml





로그 출력 및 저장

```
//로그저장
//두번째 파라미터를 넣지 않으면 name에는 dg_log가 들어감
참조 1개
public static void printLog(string contents, string name = "dg_log")
{
    DirectoryInfo di = new DirectoryInfo("dg_log");
    //if(di.Exists == false)
    if (!di.Exists)
    {
        di.Create(); //폴더 만들기
    }

    using (StreamWriter writer = new StreamWriter(@"dg_log\ " + name + ".txt", true))
    {
        writer.WriteLine(contents);
    }
}

//로그 데이터 받음
참조 23개
public static void writeLog(string contents)
{
    string logContents = $"{DateTime.Now.ToString("yyyy/MM/dd HH:mm:ss")}{contents}"; //로그내용
    string date = $"{DateTime.Now.ToString("yyyy-MM-dd")}"; //로그이름.txt

    DataManager_D.printLog(logContents, date);
}

if (dataGridView1.RowCount == 0) //도착정보가 없을때 메시지
{
    MessageBox.Show("도착정보가 없습니다.");
    DataManager_D.writeLog("도착정보가 없습니다.");
}
```

[2021_04_27].txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

[2021-04-27 15:00:25]전체 선택함
[2021-04-27 15:00:42]조회 클릭
[2021-04-27 15:00:43]출발역:부산도착역:서울
[2021-04-27 15:01:31]대구 선택함
[2021-04-27 15:19:48]전체 선택함
[2021-04-27 15:20:03]조회 클릭
[2021-04-27 15:20:03]출발역:청량리 / 도착역:안동
[2021-04-27 15:20:07]조회 클릭
[2021-04-27 15:20:07]인덱스가 범위를 벗어났습니
매개 변수 이름: index
[2021-04-27 15:20:07] 위치: System.ThrowHelp
위치: System.Collections.Generic.List`1.get_Item
위치: TrainInfo.DataManager_D.ShowInfo(String
위치: TrainInfo.InfoView_D.infoClick_Click(Object
[2021-04-27 15:29:25]대구 선택함
[2021-04-27 15:29:29]전체 선택함
[2021-04-27 15:29:35]대구 선택함
[2021-04-27 15:29:37]전체 선택함
[2021-04-27 15:30:09]대구 선택함
[2021-04-27 15:30:12]전체 선택함
[2021-04-27 15:30:25]조회 클릭
[2021-04-27 15:30:25]출발역:청량리 / 도착역:안동
[2021-04-27 15:30:29]조회 클릭
[2021-04-27 15:30:29]출발역:청량리 / 도착역:안동

프로그램에 입출력 및 에러 등의 내용을
Log화 시켜 txt 파일에 저장

시연



Q&A

