

Rad Server

오후 2:00 ~ 2:10 – 인사말

오후 2:10 ~ 3:10 – Rad Server 개요

- 환경 설정 및 서버 구축
- Rad Server 패키지 제작
- HTTP Client Application 제작

오후 3:10 ~ 3:30 - 실시간 Q&A

마이그레이션

- 워크샵: 마이그레이션을 어디서부터 시작하고 어떻게 진행하면 되는지 전반적인 마이그레이션 진행 방향을 다루는 기술 미팅(1시간)
- 프로젝트: 개발자가 직접 마이그레이션을 진행하면서 데브기어 기술 담당자에게 1:1 지원을 받을 수 있는 과정(5일)
- 컨설팅: 데브기어 기술 담당자가 마이그레이션을 리드하여 진행(계약; 기본 1달)

<https://www.devgear.co.kr/archives/products/migration-upgrade-center>

교육

- 지정 강의: 데브기어에 정해져있는 커리큘럼을 따라 일정에 맞춰 수강
- 1:1 맞춤 강의: 원하는 주제로 원하는 일정에 진행하는 1:1 맞춤형 교육

<https://www.devgear.co.kr/edu>

프로모션

- Rapid Design UX & Chart 컴포넌트 무상 제공
 - 대상: RAD스튜디오, 델파이 구매 고객
 - 엔터프라이즈 or 아키텍트 에디션
 - 기간: 2023.05.11 ~ 06.30

Rapid Design UX & Chart 란?

: 델파이 FMX 앱 개발에 사용 할 수 있는 컴포넌트 세트

<https://www.devgear.co.kr/radoffer>



세미나 발표자 : 오상현



- ❖ 엠바카데로 MVP(Most Valuable Professional) 선정 (2014년 9월)
- ❖ “한번에 개발하는 안드로이드/iOS with 델파이” 도서 공저
- ❖ 네이버 델파이 FireMonkey 개발자 카페 운영 : since 2012.04.03

<http://cafe.naver.com/delphifmx>

순서

오후 2:00 ~ 2:10 – 인사말

오후 2:10 ~ 3:10 – Rad Server 개요

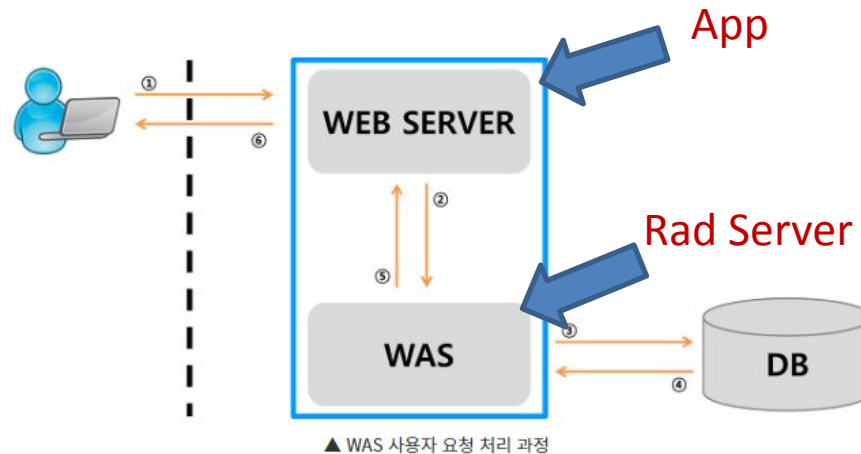
- 환경 설정 및 서버 구축
- Rad Server 패키지 제작
- HTTP Client Application 제작

오후 3:10 ~ 3:30 - 실시간 Q&A

Rad Server 개요

• 용어 정리

- **REST** : Representational State Transfer.
HTTP 를 통해 자원을 명시하고 HTTP Method(POST, GET, PUT, DELETE)를 적용.
JSON 혹은 XML를 통해 데이터를 주고 받는 것이 일반적 임.
- **WAS** : Web Application Server.



- **End Point** : API가 서버에서 리소스에 접근할 수 있도록 가능하게 하는 URL
- **EMS** : Enterprise Mobility Service가 강화된 것이다. 따라서 일부 파일이름, 컴포넌트 이름, 기술자료에서 EMS라는 용어를 만나면, RAD 서버를 의미한다고 이해해도 무방하다.

System Database



플러그인 형태의
RAD 서버 패키지(*.bpl)

RAD Server

기본 제공 리소스

- Version / API
- Users / Groups
- Installations
- Push
- Edgemodules

사용자 제작 리소스

- Resource A1
- Resource A2
- Resource B1
-

HTTP Clients



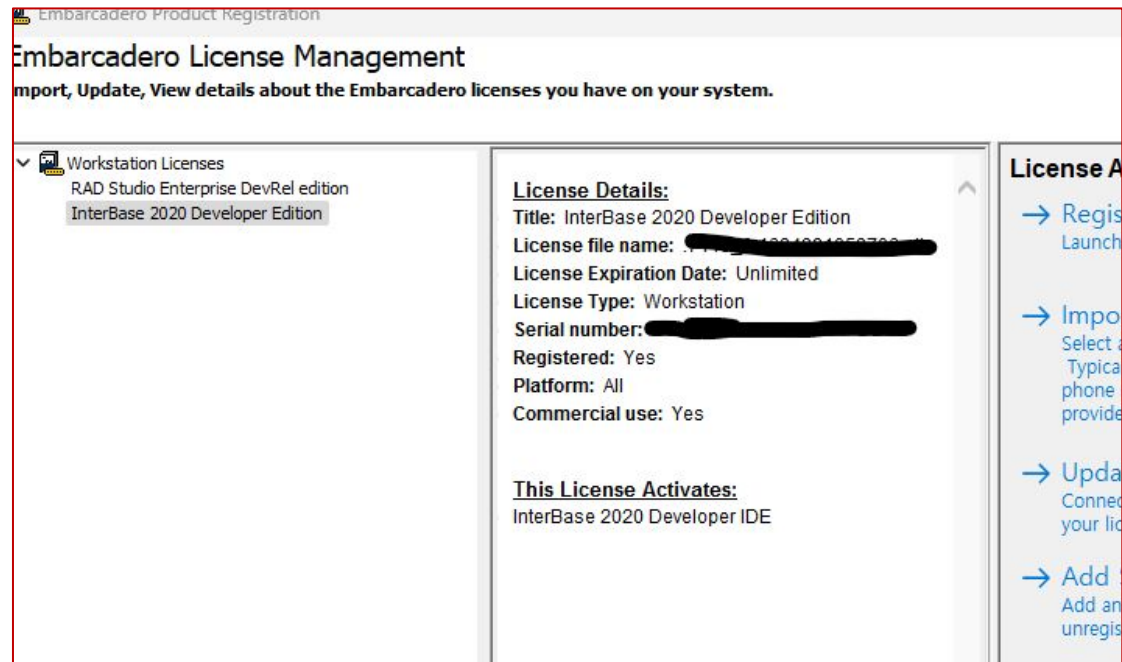
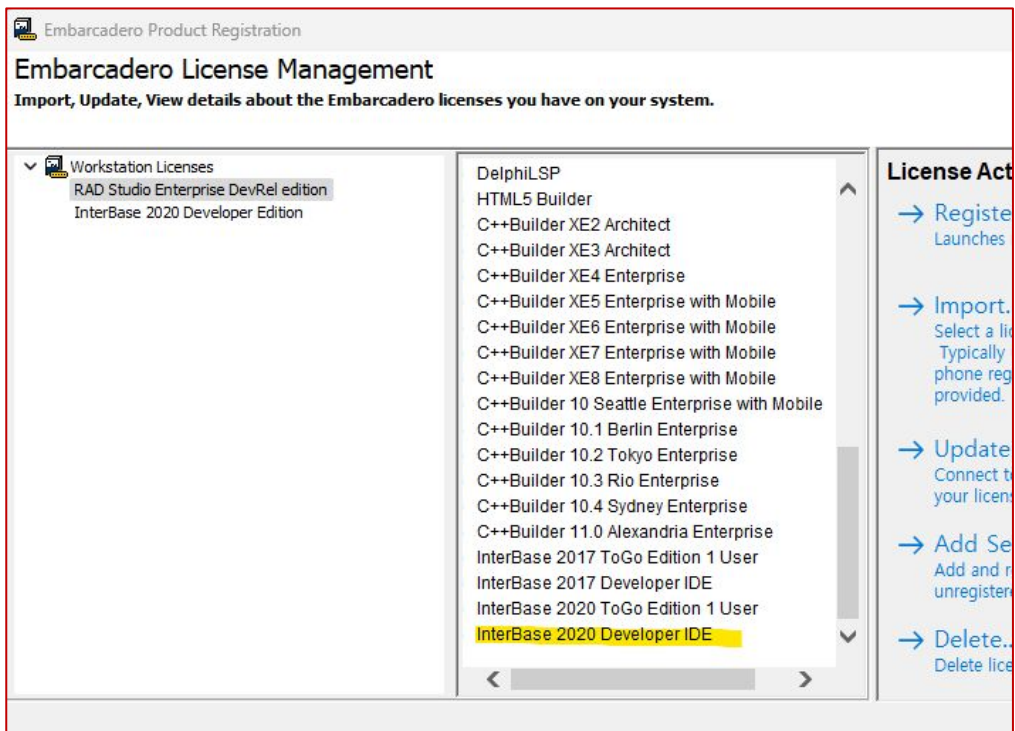
VCL / FMX



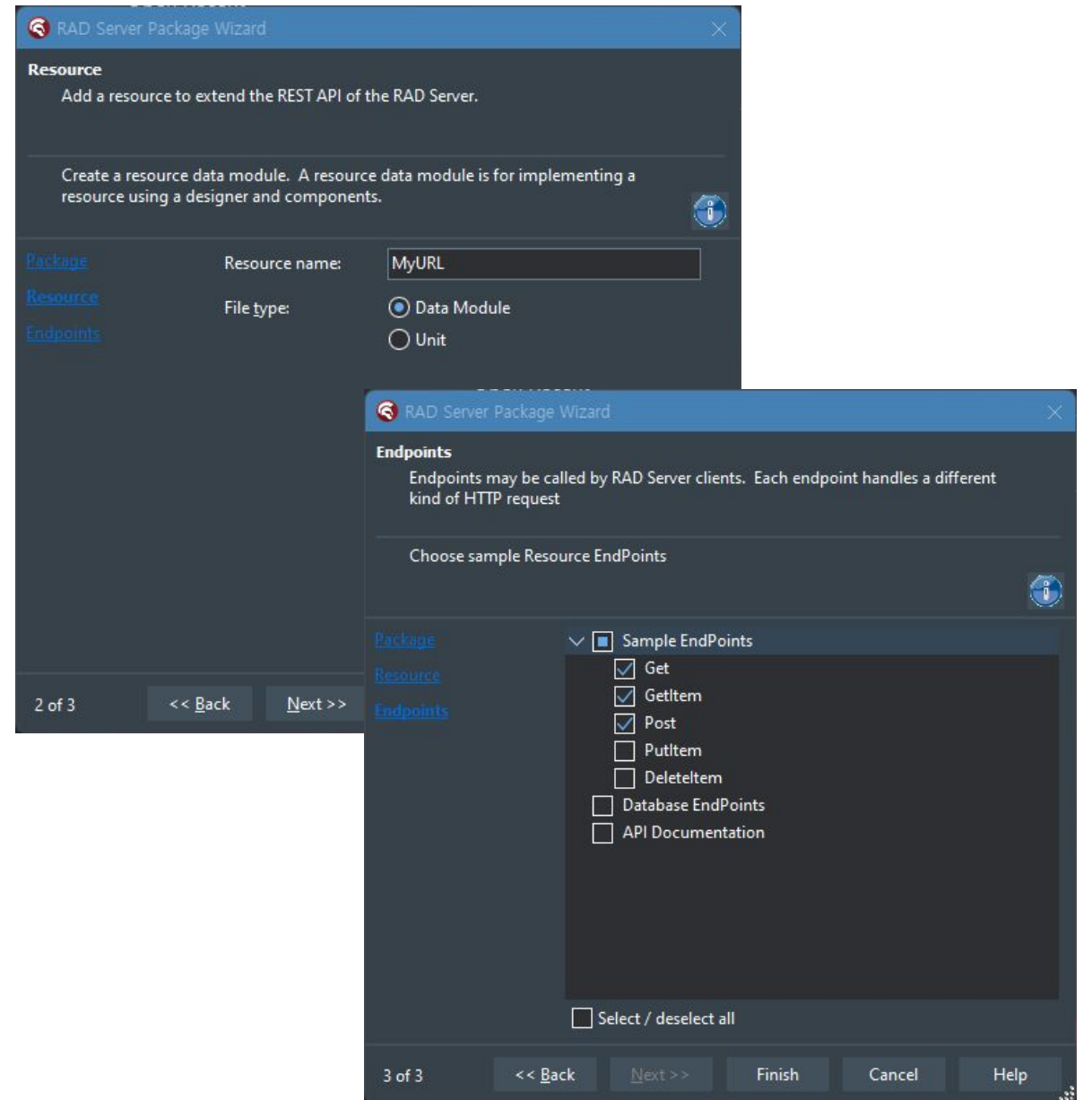
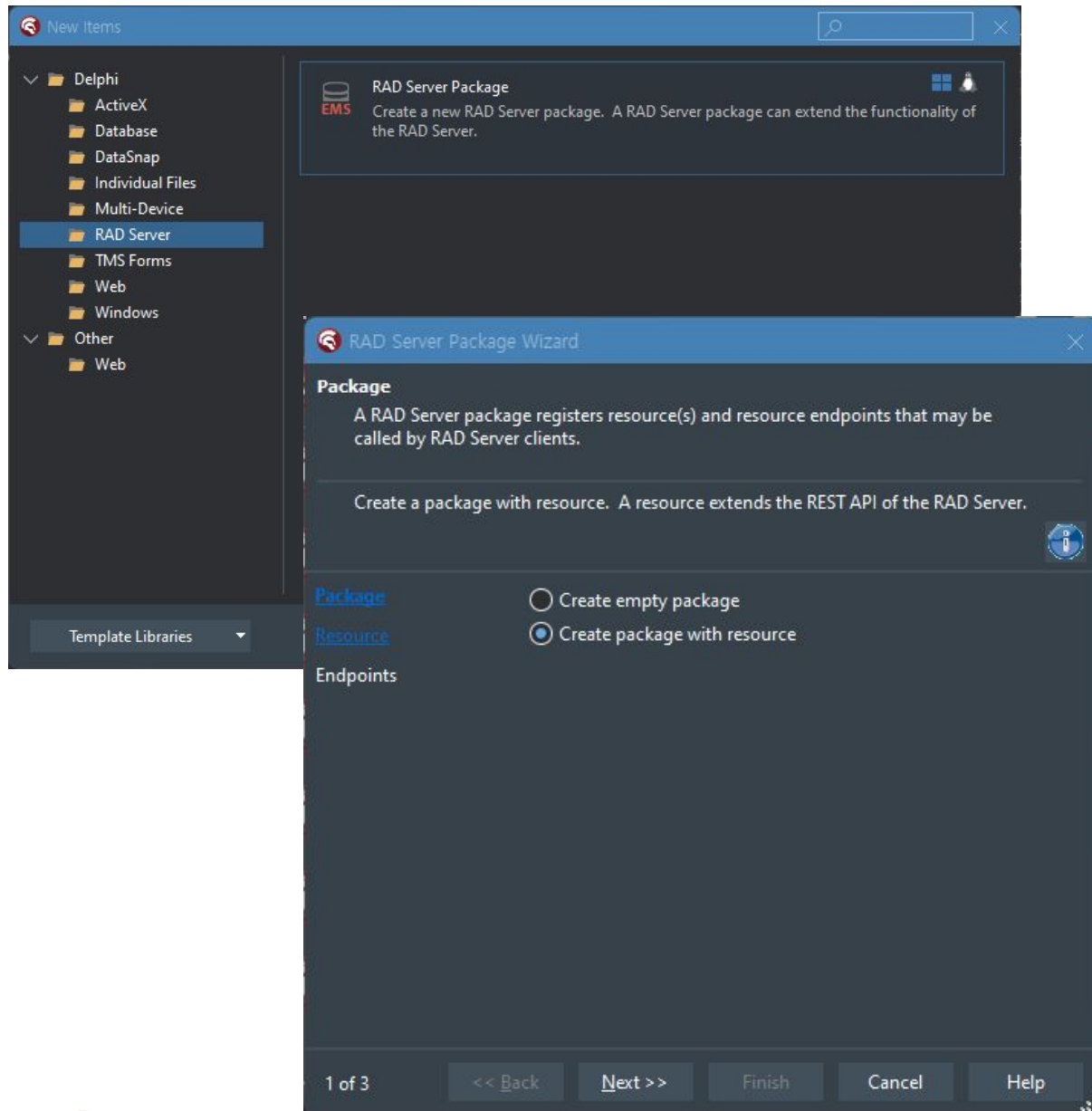
- Rad Server 특징
 - **현대화** : 기존 델파이/C++ 클라이언트/서버 비즈니스 로직을 확장 가능
 - **REST 엔드포인트 제공** : 다양한 클라이언트와 연결 가능한 범용 REST 기반 백엔드 API 제공
 - **통합 미들웨어** : RDBMS 등 외부 서버 및 서비스와 연결 가능하며, 필요한 데이터를 가공 및 제공 가능
 - **응용 프로그램 서비스** : 사용자/그룹 관리, 푸시 알림, 사용자 위치 추적 등 기본 서비스 제공
 - **개발 및 배포 용이** : RAD 서버 패키지 기반으로 개발/배포/운영 용이
 - **델파이/C++빌더 지원** : RAD 서버는 델파이와 C++빌더로 즉시 개발 가능

• InterBase 라이선스

- Rad Studio Enterprise 구매시 같이 제공 되는 InterBase 개발자용 라이선스 필요.
- InterBase 상용 라이선스 까지는 필요 없음.
- InterBase Trial 버전 라이선스도 사용 가능 ~ Trial 버전은 별도 설치 후 라이선스 매니저에 등록 해야 함

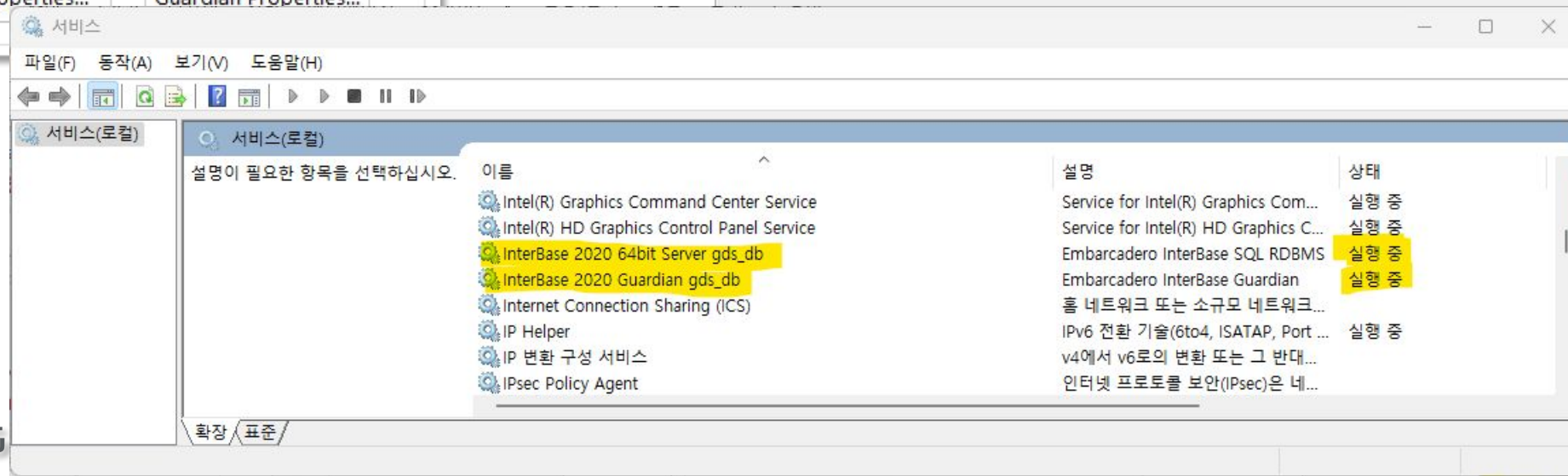
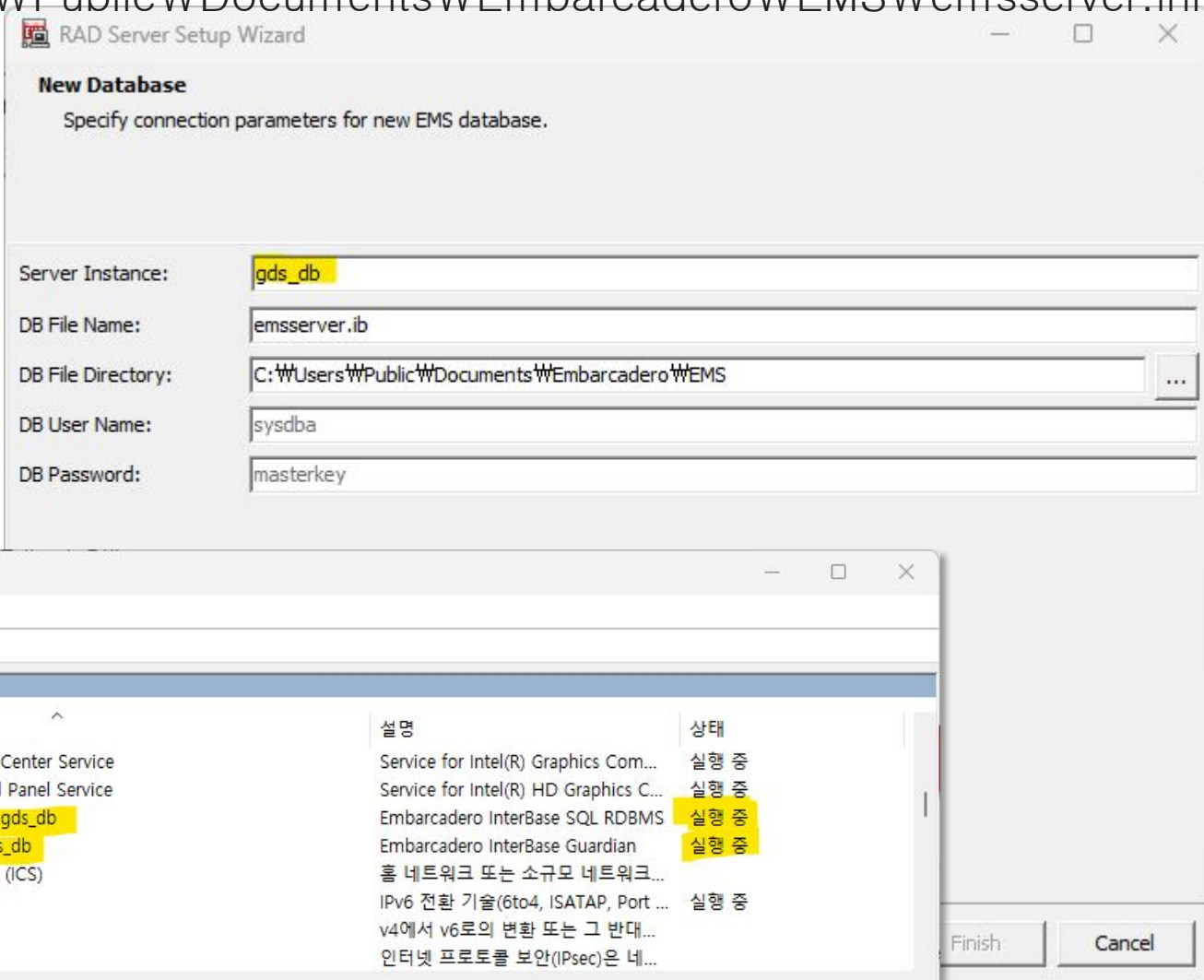
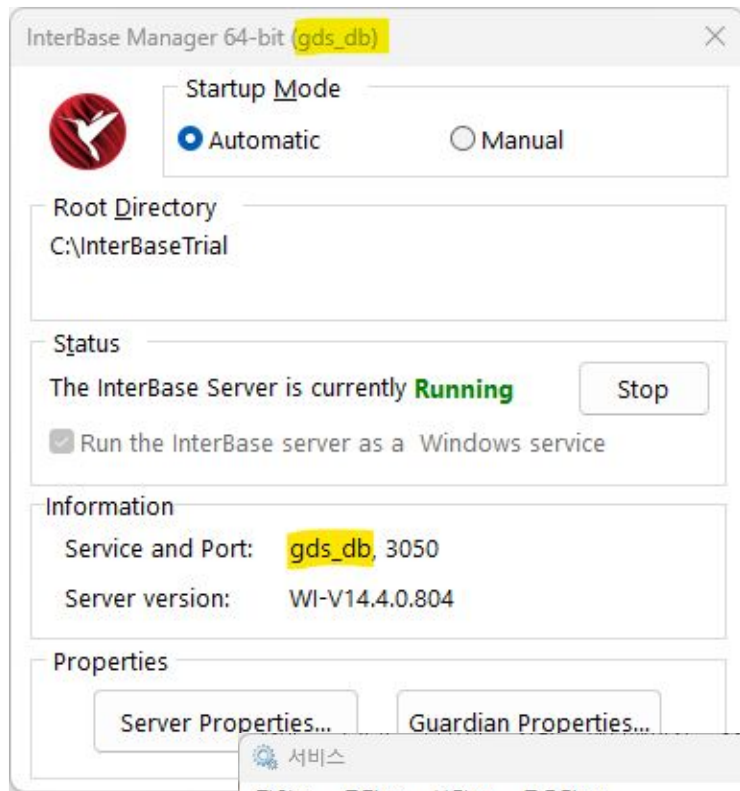


- **Rad Server 라이선스** : Rad Studio Enterprise 구매시 제공 되는 키는 배포 서버 구동 시 필요



- **emsserver.ini** 생성을 위한 최초 실행

c:\Users\Public\Documents\Embarcadero\EMS\emsserver.ini



The image displays three overlapping windows of the RAD Server Setup Wizard, illustrating the final configuration steps.

Sample Data
Choose to generate sample data into the new EMS database.

- ☒ Sample users
- ☒ Sample user groups

Console
Specify username and password to login to the EMS Console. You will need these to access the console.

Console User Name:

Console Password:

☒ Show password

Finalize
The setup wizard is ready to create files. Note that you can make changes to the emsserver.ini file later.

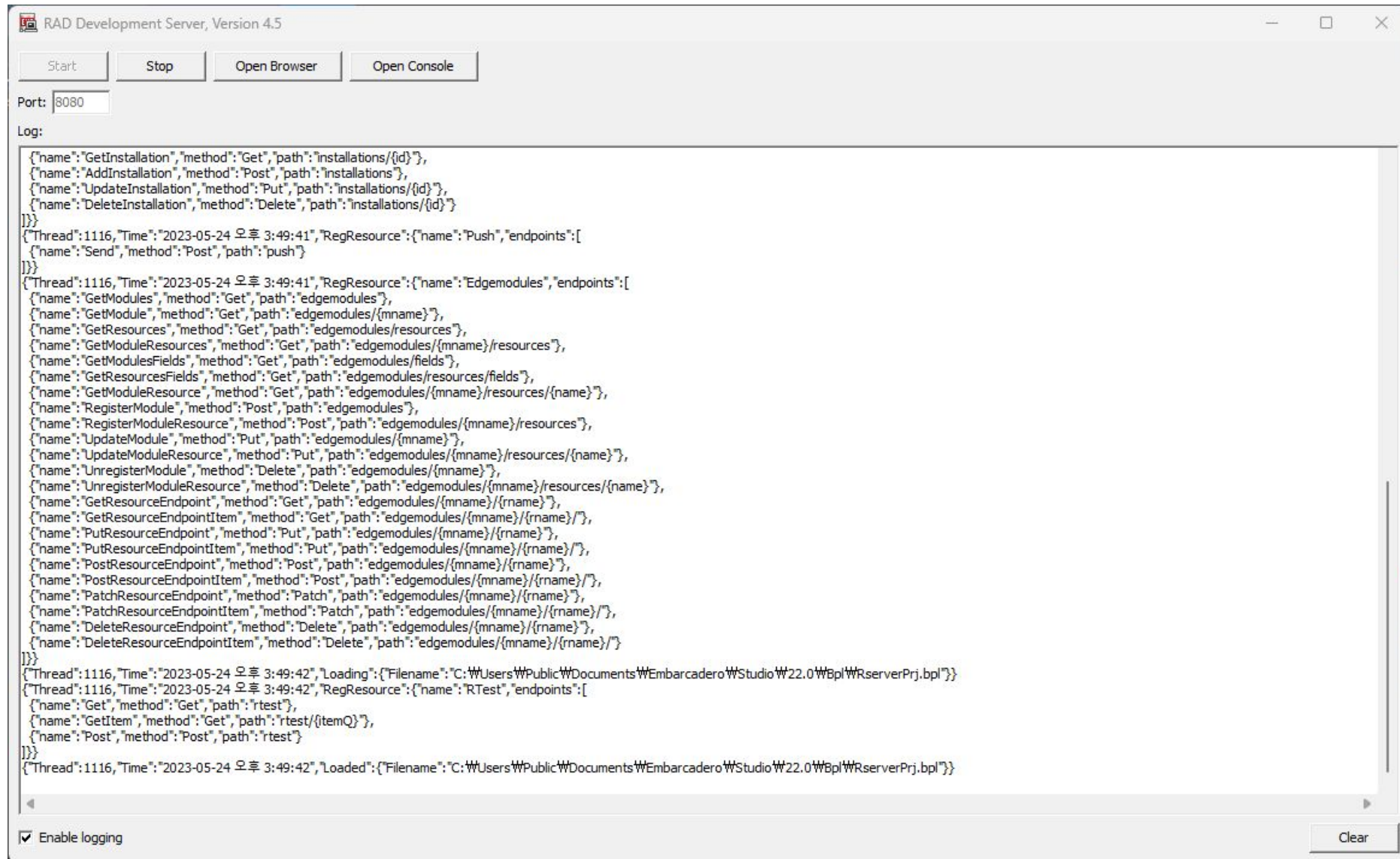
Choose "Finish" to create the following:

DB File:

Configuration File:

Registry key:

Navigation buttons: << Back, >> Next, Finish, Cancel



RAD Development Server, Version 4.5

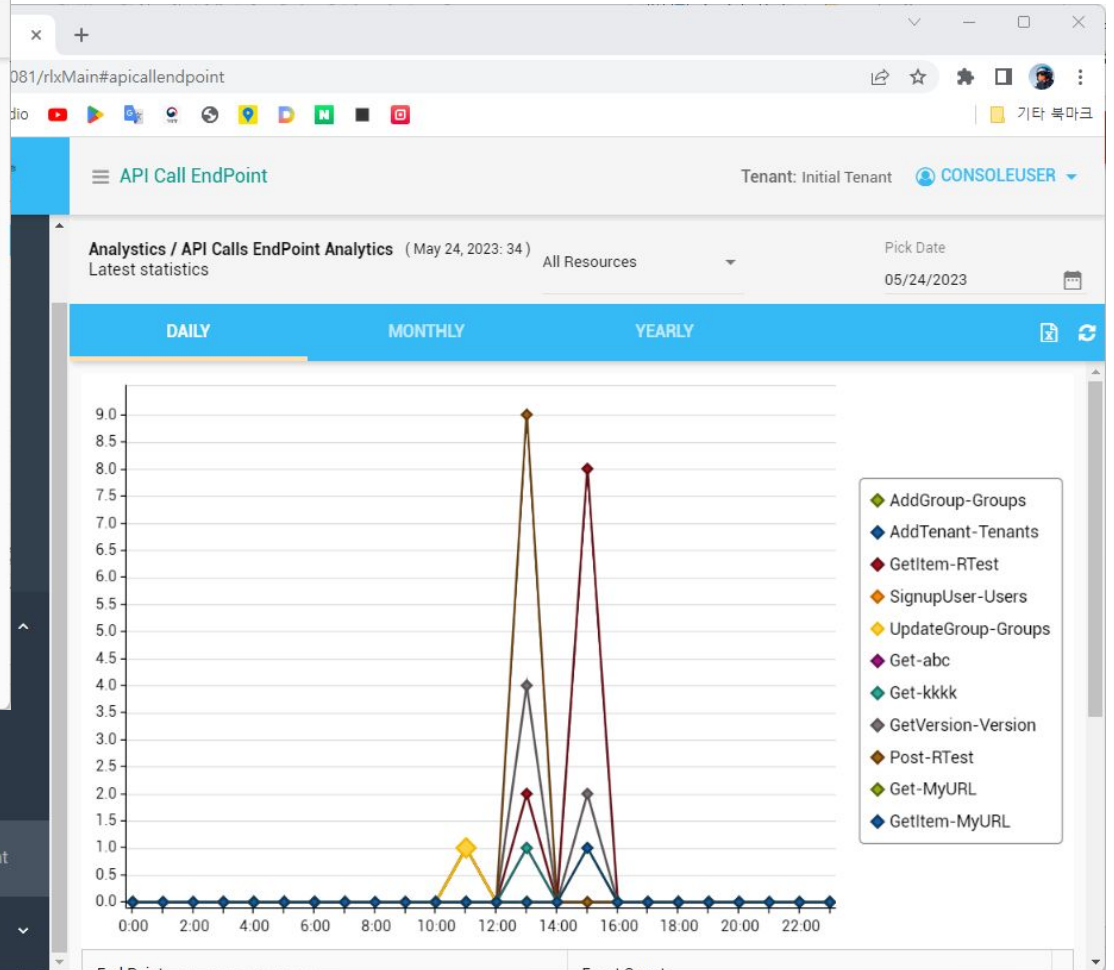
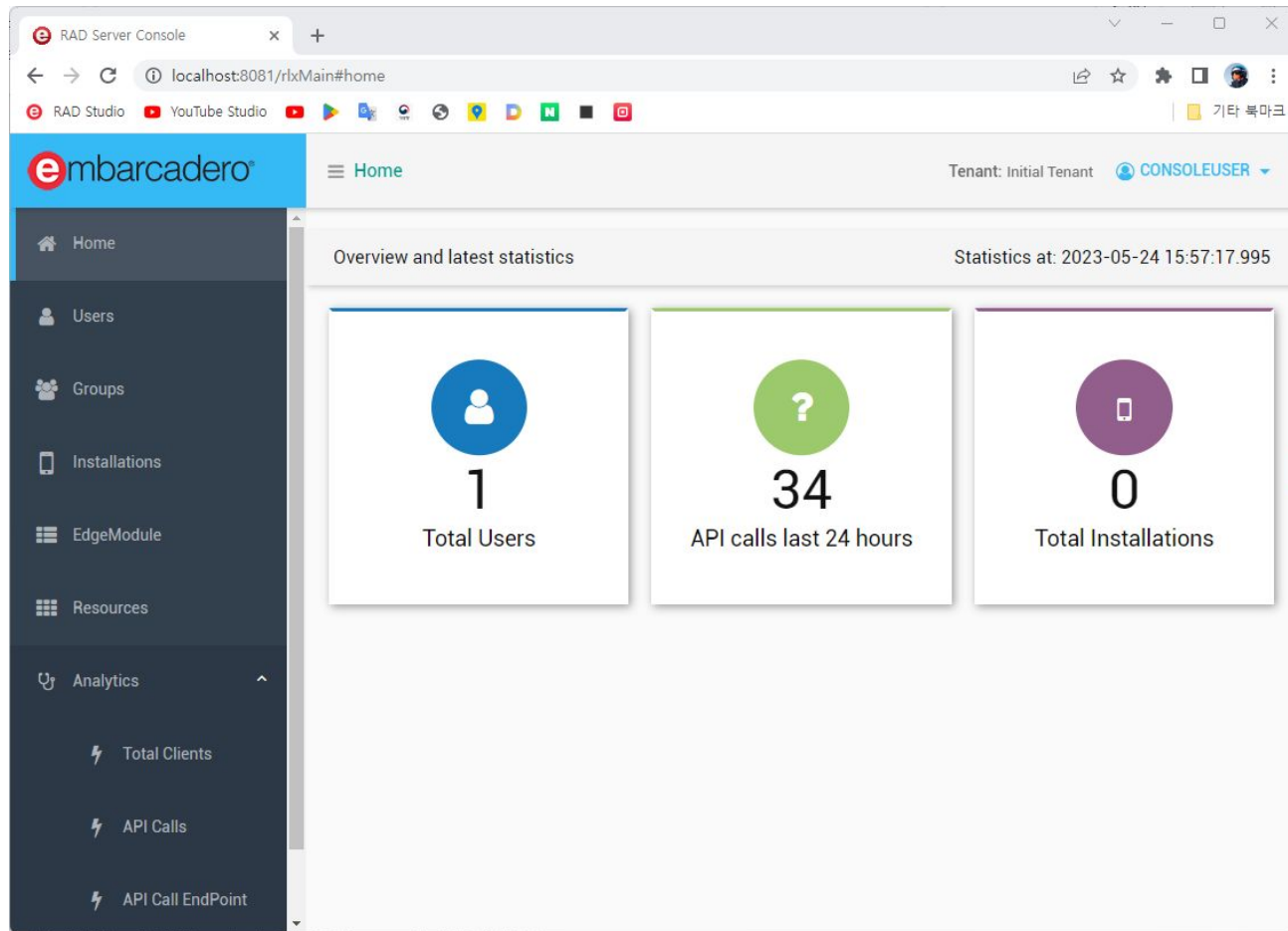
Start Stop Open Browser Open Console

Port: 8080

Log:

```
{
  "name": "GetInstallation", "method": "Get", "path": "installations/{id}",
  "name": "AddInstallation", "method": "Post", "path": "installations",
  "name": "UpdateInstallation", "method": "Put", "path": "installations/{id}",
  "name": "DeleteInstallation", "method": "Delete", "path": "installations/{id}"
}
}}
{"Thread": 1116, "Time": "2023-05-24 오후 3:49:41", "RegResource": {
  "name": "Push", "endpoints": [
    {
      "name": "Send", "method": "Post", "path": "push"
    }
  ]
}}
{"Thread": 1116, "Time": "2023-05-24 오후 3:49:41", "RegResource": {
  "name": "Edgemodules", "endpoints": [
    {
      "name": "GetModules", "method": "Get", "path": "edgemodules",
      "name": "GetModule", "method": "Get", "path": "edgemodules/{mname}",
      "name": "GetResources", "method": "Get", "path": "edgemodules/resources",
      "name": "GetModuleResources", "method": "Get", "path": "edgemodules/{mname}/resources",
      "name": "GetModulesFields", "method": "Get", "path": "edgemodules/fields",
      "name": "GetResourcesFields", "method": "Get", "path": "edgemodules/resources/fields",
      "name": "GetModuleResource", "method": "Get", "path": "edgemodules/{mname}/resources/{name}",
      "name": "RegisterModule", "method": "Post", "path": "edgemodules",
      "name": "RegisterModuleResource", "method": "Post", "path": "edgemodules/{mname}/resources",
      "name": "UpdateModule", "method": "Put", "path": "edgemodules/{mname}",
      "name": "UpdateModuleResource", "method": "Put", "path": "edgemodules/{mname}/resources/{name}",
      "name": "UnregisterModule", "method": "Delete", "path": "edgemodules/{mname}",
      "name": "UnregisterModuleResource", "method": "Delete", "path": "edgemodules/{mname}/resources/{name}",
      "name": "GetResourceEndpoint", "method": "Get", "path": "edgemodules/{mname}/{rname}",
      "name": "GetResourceEndpointItem", "method": "Get", "path": "edgemodules/{mname}/{rname}/{rname}",
      "name": "PutResourceEndpoint", "method": "Put", "path": "edgemodules/{mname}/{rname}",
      "name": "PutResourceEndpointItem", "method": "Put", "path": "edgemodules/{mname}/{rname}/{rname}",
      "name": "PostResourceEndpoint", "method": "Post", "path": "edgemodules/{mname}/{rname}",
      "name": "PostResourceEndpointItem", "method": "Post", "path": "edgemodules/{mname}/{rname}/{rname}",
      "name": "PatchResourceEndpoint", "method": "Patch", "path": "edgemodules/{mname}/{rname}",
      "name": "PatchResourceEndpointItem", "method": "Patch", "path": "edgemodules/{mname}/{rname}/{rname}",
      "name": "DeleteResourceEndpoint", "method": "Delete", "path": "edgemodules/{mname}/{rname}",
      "name": "DeleteResourceEndpointItem", "method": "Delete", "path": "edgemodules/{mname}/{rname}/{rname}"
    }
  ]
}}
{"Thread": 1116, "Time": "2023-05-24 오후 3:49:42", "Loading": {
  "Filename": "C:\\Users\\Public\\Documents\\Embarcadero\\Studio\\22.0\\Bpl\\RserverPrj.bpl"
}}
{"Thread": 1116, "Time": "2023-05-24 오후 3:49:42", "RegResource": {
  "name": "RTest", "endpoints": [
    {
      "name": "Get", "method": "Get", "path": "rtest",
      "name": "GetItem", "method": "Get", "path": "rtest/{itemQ}"
    },
    {
      "name": "Post", "method": "Post", "path": "rtest"
    }
  ]
}}
{"Thread": 1116, "Time": "2023-05-24 오후 3:49:42", "Loaded": {
  "Filename": "C:\\Users\\Public\\Documents\\Embarcadero\\Studio\\22.0\\Bpl\\RserverPrj.bpl"
}}
```

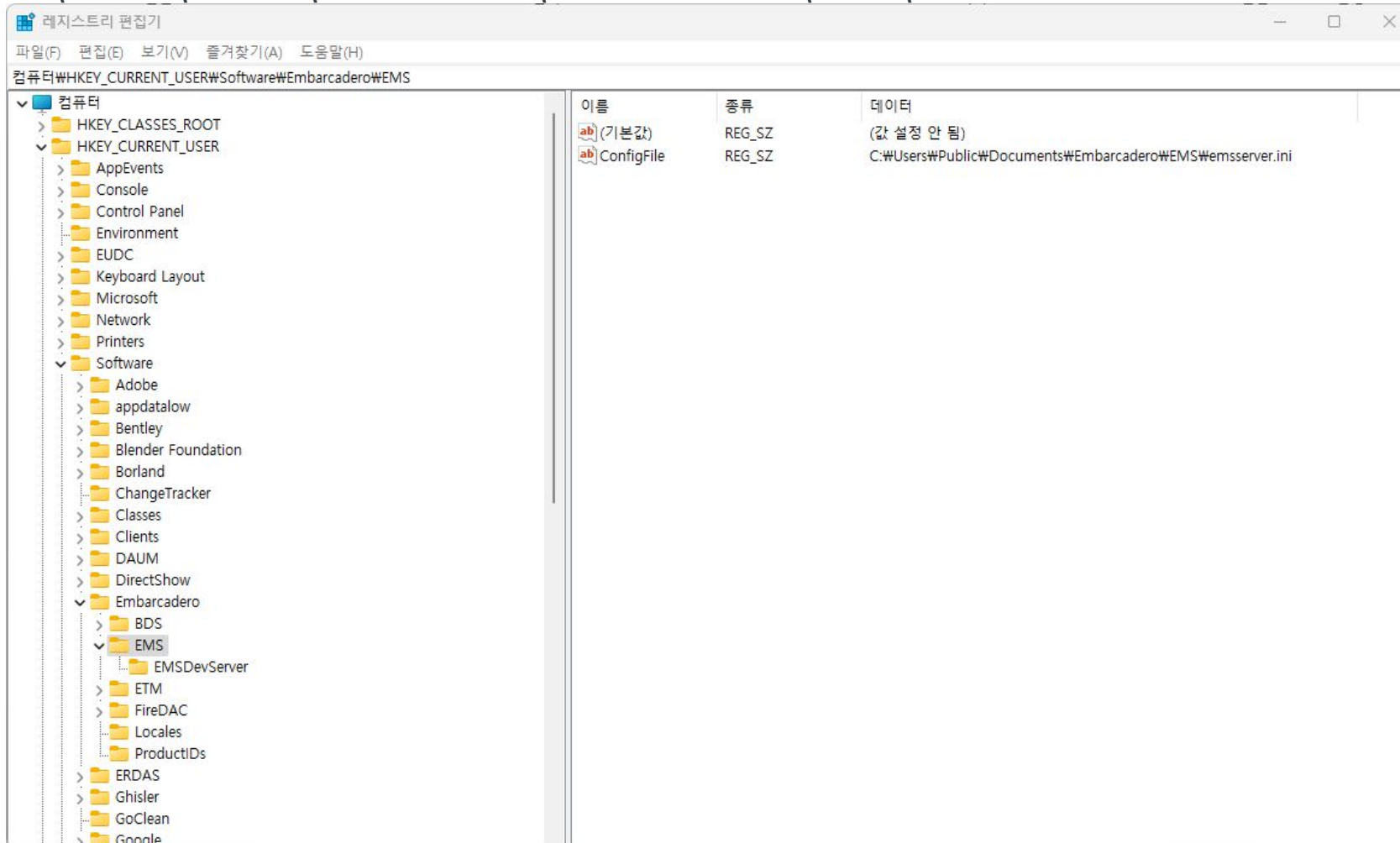
☒ Enable logging Clear



emsserver.ini 파일이 없거나 경로를 못 찾는 오류 메시지 : 레지스트리 편집기
항목 추가

HKEY_CURRENT_USER\Software\Embarcadero\EMS 경로에 **ConfigFile** 설정

Ex) C:\Users\Public\Documents\Embarcadero\EMS\emsserver.ini

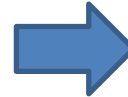


Rad Server HTTP Client App 제작

2 Tier Case

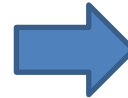
- **DB Client App**

Connection Method



- **Json REST App**

SQL Data logic

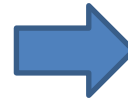


- **Rad Server**

3 Tier Case

- **Client App**

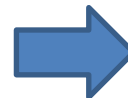
Connection Method



- **Json REST App**

- **M/W Server**

SQL Data logic



- **Rad Server**

DB SQL 조회 값을 Json 으로 저장 하는 법 예시 (1/2)

DB Browser for SQLite - D:\5sh_Delphi\Seminar2023\3.RadServer\Server\test.db

파일(F) 편집(E) 보기(V) 도구(T) 도움말(H)

새 데이터베이스(N) 데이터베이스 열기(O) 변경사항 저장하기(W) 변경사항 취소(C)

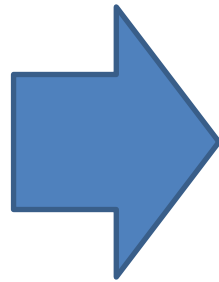
데이터베이스 구조 데이터 보기 Pragma 수정 SQL 실행

테이블(T): table1

	inc	field1	field2	field3	field4
	...	필터	필터	필터	필터
1	1	경기도	안산시	대부동	대부도
2	2	경기도	안산시	대부동	불탄도
3	3	경기도	안산시	대부동	선감도
4	4	경기도	안산시	대부동	육도
5	5	경기도	안산시	대부동	풍도
6	6	경기도	화성군	서신면	제부도
7	7	경기도	화성군	송산면	어도
8	8	경기도	화성군	송산면	우음도
9	9	경기도	화성군	송산면	형도
10	10	경기도	화성군	우정면	국화도
11	11	경상남도	거제시	거제면	산달도
12	12	경상남도	거제시	둔덕면	화도
13	13	경상남도	거제시	사등면	가조도
14	14	경상남도	거제시	사등면	고개도
15	15	경상남도	거제시	유인도	거제도
16	16	경상남도	거제시	일운면	내도
17	17	경상남도	거제시	일운면	외도
18	18	경상남도	거제시	일운면	지심도
19	19	경상남도	거제시	장목면	이수도
20	20	경상남도	거제시	하청면	철천도
21	21	경상남도	거제시	하청면	황덕도
22	22	경상남도	고성군	삼산면	와도

1 - 22 of 507

특정 레코드 행으로 가기: 1



```
127.0.0.1:8080/RTest/경기도 x +
127.0.0.1:8080/RTest/경기도
RAD Studio YouTube Studio
// 20230530134121
// http://127.0.0.1:8080/RTest/%EA%B2%BD%EA%B8%B0%EB%8F%84
{
  "Count": 10,
  "MyItems": [
    {
      "inc": 1,
      "field1": "경기도",
      "field2": "안산시",
      "field3": "대부동",
      "field4": "대부도"
    },
    {
      "inc": 2,
      "field1": "경기도",
      "field2": "안산시",
      "field3": "대부동",
      "field4": "불탄도"
    },
    {
      "inc": 3,
      "field1": "경기도",
      "field2": "안산시",
      "field3": "대부동",
      "field4": "선감도"
    },
    {
      "inc": 4,
      "field1": "경기도",
      "field2": "안산시",
      "field3": "대부동",
      "field4": "육도"
    }
  ]
}
```

DB SQL 조회 값을 Json 으로 저장 하는 법 예시 (2/2)

```
function TDataModule1.Select_All() : SAnsType;
begin
    result.sFd1 := TStringList.Create;
    result.sFd2 := TStringList.Create;
    result.sFd3 := TStringList.Create;
    result.sFd4 := TStringList.Create;

    FDConnection1.Open;
    try
        FDQuery1.Close;
        FDQuery1.SQL.Clear;
        FDQuery1.SQL.Add( 'Select * from table1' );
        FDQuery1.Open;
        FDQuery1.First;

        // 조회한 데이터를 레코드에 저장.
        while Not FDQuery1.EOF do
            begin
                result.sFd1.Add( FDQuery1.FieldByName('field1').AsString );
                result.sFd2.Add( FDQuery1.FieldByName('field2').AsString );
                result.sFd3.Add( FDQuery1.FieldByName('field3').AsString );
                result.sFd4.Add( FDQuery1.FieldByName('field4').AsString );

                FDQuery1.Next;
                Inc( result.qCount );
            end;
            result.qCount := result.sFd1.Count;
        except
            on e: Exception do begin
                ShowMessage( e.Message );
            end;
        end;

        FDConnection1.Close;
    end;
```



```
function TDataModule1.A_Query: string;
var
    JTopObj, JsubObj : TJSONObject;
    JArr : TJSONArray;
    JPair : TJSONPair;
begin
    JTopObj := TJSONObject.Create; // 메인 블록 정의

    try
        FDConnection1.Open;
        try
            FDQuery1.Close;
            FDQuery1.SQL.Clear;
            FDQuery1.SQL.Add( 'Select * from table1' );
            FDQuery1.Open;
            FDQuery1.First;

            JArr := TJSONArray.Create; // Json 배열 정의
            while Not FDQuery1.EOF do
                begin
                    JsubObj := TJSONObject.Create; // 각 항목 블록

                    JsubObj.AddPair( 'field1', FDQuery1.FieldByName('field1').AsString );
                    JsubObj.AddPair( 'field2', FDQuery1.FieldByName('field2').AsString );
                    JsubObj.AddPair( 'field3', FDQuery1.FieldByName('field3').AsString );
                    JsubObj.AddPair( 'field4', FDQuery1.FieldByName('field4').AsString );

                    JArr.AddElement( JsubObj );
                    FDQuery1.Next;
                end;

            JPair := TJSONPair.Create( 'Items', JArr ); // Items 항목 아래에 배열세트를 넣는다.
            JTopObj.AddPair( 'Count', TJSONNumber.Create( FDQuery1.RecordCount ) ); // 상단에 레코드 카운트 표시
            JTopObj.AddPair( JPair ); // 메인블록에 Items 배열세트 넣음.

        except
            on e: Exception do begin
                result := e.Message;
            end;
        end;

    finally
        FDConnection1.Close;

        result := JTopObj.ToString; // 결과값 전달
        JTopObj.Free;
    end;
end;
```

- 데모순서

1. 위저드로 Rad Server 만들고 실행 및 브라우저 접속 확인
2. Rad Server에 FireDAC 탑재
3. 크롬 브라우저 활용 테스트
4. Client App 사용
5. 엠바카데로 콘솔 앱 보기

- 추가 자료

- Docwiki 문서 : [https://docwiki.embarcadero.com/RADStudio/en/RAD_Server_\(EMS\)](https://docwiki.embarcadero.com/RADStudio/en/RAD_Server_(EMS))
- Rad Server 배포
 - 윈도우 서버 : [RAD 서버를 윈도우 서버 2019 IIS 10 환경에 설치하기](#)
 - 리눅스 서버 : [How-To: Deploy The Production Version Of RAD Server To Linux](#)
 - Rad Server Lite : [RSLite\(배포/설치가 간소화된 RAD 서버\) 배포하기](#)
 - [RAD Server Docker Deployment](#)

Q & A



[주관식입니다]

API가 서버에서 리소스에 접근 할 수 있도록 하는
URL 을 호칭하는 용어는 ?



정답!

End point (엔드포인트)



[주관식입니다]

RAD서버로 프로젝트를 빌드 했을 때
생성되는 결과물의 파일 확장자는 ?



정답!

bpl



[주관식입니다]

RAD서버로 프로젝트의 속성을
저장하는 파일명은 ?



정답!

emsserver.ini



- **데브기어 기술 포럼 (welcome.devgear.co.kr)**

- 유용한 기술 콘텐츠들이 가장 빠르게 업데이트됩니다.
- 기술, 설치/등록 Q&A도 여기를 이용하세요!
- 소중한 지식과 경험을 나누어주신 매 월 '이 달의 회원' 선정

- **데브기어 RAD서버 관련 콘텐츠**

- 기술 포럼: <https://welcome.devgear.co.kr/tags/rad%20서버/>



• 파이썬 GUI Apps Con 2023 (2023.6.14~6.15)

- 엠바카데로와 저명한 업계 전문가들과 함께 DelphiFMX, PySimpleGUI, Tkinter 등등에 대한 이야기를 나눠봅시다!
 - 다루는 내용: 파이썬 GUI 프레임워크, 파이썬 그래픽 GUI, 데스크탑 UX 최적화 및 구축 전략, DelphiFMX와 AI 언어 모델 등등
- 신청하기: pythongui.org/python-gui-apps-con-2023/



• 개발 사례

• 엔터프라이즈 소프트웨어 개발 사례

- 34밀리초 안에 암호화 처리해 디지털 데이터 보호
- 델파이로 완성한 코어 बैं킹 시스템
- 비주얼 C++을 버리고 델파이로 전환해 60배의 놀라운 속도 향상 달성 방법 등등
- 전체 보기: welcome.devgear.co.kr/topic/1059-엔터프라이즈-소프트웨어-개발-사례-다양한-사례들을-확인해보세요/

•

• 데브기어 무상 지원 서비스

- **보도자료 배포** - 작성부터 배포까지 모두 데브기어에서 지원
- **데브기어 홈페이지, SNS 등에 게재**
- 신청: www.devgear.co.kr/casestudy/reg

