LinqPad for DevOps

개발도 모자라서 운영 대응도 해야 하는
~~불쌍한~~ C# 엔지니어를 위하여

2016.04.30 Bob

MOBIL FACTORY

이 자료는

- **가볍고 짱 쎈 LinqPad 소개**
- C#도 스크립트 처럼 가볍게 써 봅시다
- DevOps 라는 거창한 제목으로 낚시

- 저 LinqPad ReSeller 아닙니다 ~~(아직은)~~

목차

- LinqPad가 왜 필요한지 설명
- LinqPad가 어디가 좋은지 설명
- LinqPad를 어떻게 쓰고 있는지 설명
- LinqPad를 더 잘 쓰기 위한 방법 소개
- 부록, Scirpt 작성 tip

Live operation…

Live service의 대응

- **미리 대비하기 어려운** 요청이 다양하게 발생
  - 이상한 조건의 이벤트 대상자 추출 해 주세요!
  - 특정 조건의 유저들 데이터 이상한 것 같아요!
  - 보상 지급 잘못 된 것 같아요!
  - ~~예상치 못한~~ 유저 통계 뽑아주세요!
  - 전체 유저 json 데이터에서 A를 B로 바꿔주세요!
  - 랜덤 로직 제대로 동작하는 것 맞아요?
  - 테스트 코드 돌려보고 싶어요 !

Live service의 대응

운영 스크립트 작업의 특징

- 급하게 만드는 **임시 스크립트**
- 한 번 쓰고 ~~아마도~~ 다시 안 쓸 코드
  - 재사용이 필요하면 스크립트로 때우지 말고 tool 을 만듭시다.
- 서버나 게임 코드를 참조해야 함
  - 보통 User class 나 logic helper class
- DB에 접속해서 데이터를 추출하기도 함

운영 스크립트 작업의 딜레마

- Q. C# 스크립트 작업을 **어디서** 하면 될까요?

- A1. **새 프로젝트** 파면 되지
  - 기존 class 참조 할 일이 너무 많아요
  - user class만 가져오고 싶은데 참조가 너무 많아요
  - Helper class 복사 붙여 넣기 하는데 구조가 이상해요

운영 스크립트 작업의 딜레마

- Q. C# 스크립트 작업을 **어디서** 하면 될까요?

- A2. **기존 프로젝트**에 넣으면 되지
  - 너무 무거움
  - 운영 이슈로 사소 수정이 너무 잦음
  - API 구조 변경이나 class rename 할 때
    다시는 안 쓰는 운영 스크립트도 수정해 줘야 해

운영 스크립트 작업의 딜레마

- Q. C# 스크립트 작업을 **어디서** 하면 될까요?
- A3. 그럼 그냥 쓰고 지워야 겠네

운영 스크립트 작업의 딜레마

LinqPad !

C# 쓴다면, Linqpad 쓰세요. 두 번 쓰세요

LINQPad

- 단 하나의 C# 파일
- F5 누르면 바로 실행 !
- .Dump() 하면 뭐든지 알아서 잘 출력

LINQPad

- 단 하나의 C# 파일
- **임시코드를 작성하기에 최적!**
- F5 누르면 바로 실행 !

- .Dump() 하면 뭐든지 알아서 잘 출력

LINQPad

- 단 하나의 C# 파일
- **임시코드를 작성하기에 최적!**
- F5 누르면 바로 실행 !
- **가볍고 빠르게 뜬다!**
- .Dump() 하면 뭐든지 알아서 잘 출력

LINQPad

- 단 하나의 C# 파일
- **임시코드를 작성하기에 최적!**
- F5 누르면 바로 실행 !
- **가볍고 빠르게 뜬다!**
- .Dump() 하면 뭐든지 알아서 잘 출력
- **Class 내부 변수 확인하기에 매우 편리!**

LINQPad – 단 하나의 파일

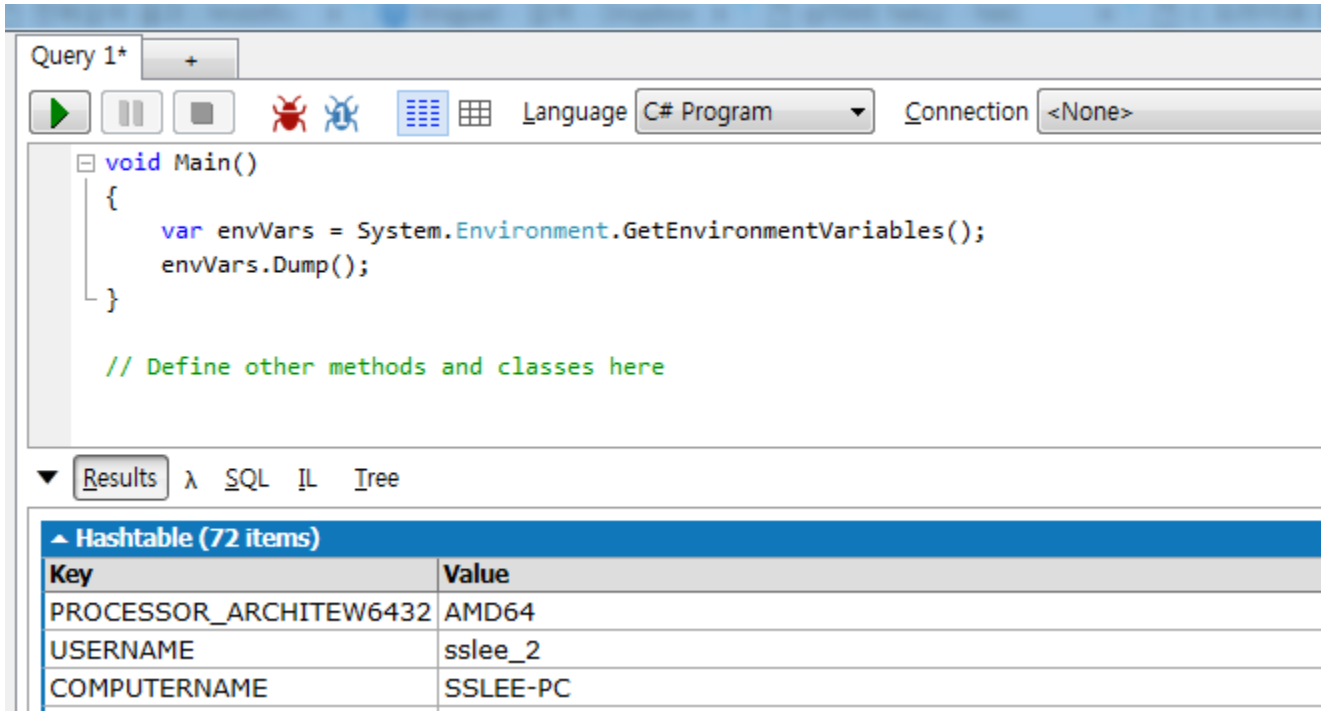
- 임시 코드를 위해 새로 프로젝트를 만들거나
- 스크립팅 하는 데 프로젝트 빌드를 기다리거나
- 서비스 실행 시간을 기다릴 필요가 없어요

- F5 만 누르면 바로 실행됩니다.

LINQPad 실행 예시

Code ->

Result ->



LINQPad – dll 참조하기

- 테스트 코드에서 user class 같은걸 임시로 만들 필요가 없어요
- 기존 프로젝트의 코드를 **직접 참조** 가능합니다
- 기존 프로젝트 dll을 넣기만 하면 끝

LINQPad – dll 참조하기

- DLL 참조하는 방법
  1. 기존 프로젝트에서 dll을 빌드 해서
  2. dll을 linqpad Plugin 경로에 추가

LINQPad – dll 참조하기

- DLL 참조하는 방법
  1. 기존 프로젝트에서 dll을 빌드 해서
     (Visual Studio \Debug 가면 있는 dll을 전부 복사)
  2. **dll을 linqpad Plugin 경로에 추가**
     (linqpad 경로 변경 단축키 f4)

LINQPad – Dump() 사용

- 변수 출력할 때
  - 변수 type 신경 쓸 필요가 없어요.
  - Collection Foreach를 할 필요도 없어요.
  - Null exception 걱정할 필요도 없어요

- List, Dictionary, struct, class 뭐든지
- **.Dump() 만 찍어주면 바로 출력** 됩니다.

Dump() 예시

- 적당히 접고 펴는 것도 잘 됩니다

- 참고로 단순 출력은 Console.WriteLine()

```
        // 01 Read
        var user = await UserHelper.GetUser("full01");
        user.BoardQuest.Dump();
    }
```

Results λ SQL IL Tree

Dictionary<String,BoardQuestModel> (9 items)

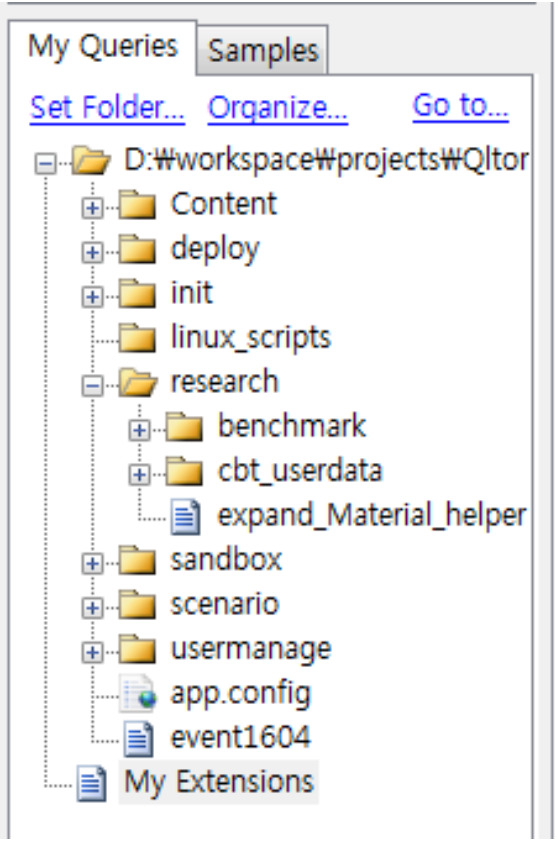
Key	Value
0	BoardQuestModel
	Qlton.Model.BoardQuestModel
	NameKey1: 10210
	NameKey2: 10070
	Orders: List<BoardQuestOrderModel> (1 item)
	DataId: 20700010, Count: 1
	DeletedAt: 0
	RegenTime: 0
	RewardGem: 14
	RewardExp: 6
	Next: *null*
1	BoardQuestModel
	Qlton.Model.BoardQuestModel
	NameKey1: 10320
	NameKey2: 10050

파일간 독립성 보장

- 개별 **파일마다 따로 빌드 & 실행**

- 같이 참조하는 코드 수정했더니,
  옛날에 만든 다른 파일에서 컴파일 오류 남.
- **그냥 무시할 수 있음!**

- **안 쓰는 코드는 유지보수 할 필요 없어짐**
- 또 쓰는 코드는 필요할 때 수정하면 됨



이 모든 게 무료 !!

~~(하지만, 일부 기능은 유료입니다)~~

지금 linqpad 다운로드 하세요

이런 문서 100번 보는 것보다
한 번 써보면 이해가 더 잘 됨

LinqPad 사용 사례

LinqPad 어디에 쓰고 있나요?

사용자 서비스 프로젝트에

직접 포함 될 필요가 없는

모든 작업에 사용 가능

LinqPad 어디에 쓰고 있나요?

- EX1. 유저 관리 스크립트
- EX2. 코드 기능 테스트
- EX3. 개발자가 사용할 툴 서비스 제작
- Ex4. 전체 유저 작업 및 임시 통계

LinqPad 어디에 쓰고 있나요?

- Ex1. 유저 관리 스크립트
  - 툴 구현 없이, 모든 임시 작업 가능

  - 유저 정보 임시 조회, 수정, 테스트 계정 생성
  - 유저 상태 백업 및 복구
  - CS 대응. 실 서버에서 개발 서버로 유저 상태 복제
  - 전체 유저 목록 csv 추출
  - 전체 유저 데이터 수정 일괄 처리

LinqPad 어디에 쓰고 있나요?

- Ex2. 기능 구현 테스트
  - **클라 구현 & 빌드 없이 모든 기능 테스트 가능**

  - 서버 API 변경 및 하위호환 테스트
  - 랜덤 로직 반복 수행 및 확인
  - 기획 공식 변경 시, 코드 시뮬레이션 및 결과 확인
  - 서버 스트레스 테스트 코드

LinqPad 어디에 쓰고 있나요?

- Ex3. 개발자가 사용할 툴 서비스 제작
  - 프로그래머만 쓰는 tool은 linqpad로 작성 한다

  - Task Sceduler로 주기적인 작업 처리
  - 패치 배포 용 스크립트
    - 서버 코드 빌드 수행
    - AWS SDK로 빌드 결과를 업로드
    - 명령을 받을 서비스 nancy 로 구현
    - Nancy가 명령을 받으면 빌드 결과 다운하고 서버에 적용

LinqPad 어디에 쓰고 있나요?

- Ex4. 전체 유저 작업 및 임시 통계
  - **서비스 전역에서 해당하는 문제 일괄 처리**

    - 특정 조건에 맞는 유저 목록 추출
    - 유저 데이터 일괄 수정
    - 임시 데이터 통계 추출 및 현재 상태 파악

LinqPad 어디에 쓰고 있나요?

사용자 서비스 프로젝트에
직접 포함 될 필요가 없는 모든 작업

- EX1. 유저 관리 스크립트
- EX2. 코드 기능 테스트
- EX3. 개발자가 사용할 툴 서비스 제작
- Ex4. 전체 유저 작업 및 임시 통계

LinqPad 추가 기능

MyExtension

- 단일 파일이라 힘들어요 ?!
- LinqPad 여러 파일에서 공통으로 쓰고 싶은 경우 사용

- **LinqPad 전체에서 같은 코드 참조 가능**
  - MyExtenson.linq 안에 구현해야 함
  - Helper 함수/클래스 추가 가능

  - (SeeAlso) this keyword 로 기존 class 에 method binding 도 가능
    (https://msdn.microsoft.com/en-us/library/cc981895.aspx)

LinqPad CLI

- Linqpad 클릭 없이 실행하고 싶을 때
  - .Linq 코드 작성 후 해당 파일을 CLI로 실행 가능

- Window Task Manager에서 실행하도록 연동 가능
- Thread 생성도 가능
- Nancy 등을 연동하여 서비스 형태로도 실행 가능

여기 까지도 무료 !!

~~(일부 기능은 유료입니다)~~

Linqpad product

이제서야 유료 기능 소개

LinqPad 핵심 유료 기능

- **무료 버전으로도 사용하는데 무리 없음**

- 코드 자동 완성 (pro 부터)
- Nuget 전체 검색 지원 (Developer 부터)
- Breakpoint debugging (Premium 부터)

Purchase

https://www.linqpad.net/Purchase.aspx

	EDITION			
	FREE	PRO	DEVELOPER	PREMIUM
Run LINQ queries - or any C#/VB/F# expression/program	✔	✔	✔	✔
Results to rich text or data grids	✔	✔	✔	✔
Automatic translation to SQL, fluent-lambda and IL	✔	✔	✔	✔
Reference your own assemblies, plus the entire .NET Framework	✔	✔	✔	✔
Use custom extensions, visualizers and data source providers	✔	✔	✔	✔
Full C#/F#/VB autocompletion and tooltips		✔	✔	✔
Smart tags for importing additional namespaces/references (C#/VB)		✔	✔	✔
Code outlining and auto-formatting (C#/VB)		✔	✔	✔
Rename symbol, go to definition and find references (C#/VB)		✔	✔	✔
Built-in and custom code snippets			✔	✔
Cross-database querying for SQL Server			✔	✔
Directly edit SQL data in grids and save changes back to database			✔	✔
Full NuGet integration			✔	✔
Integrated debugger, with call stack, threads, local variables/watch windows, breakpoints and single-stepping				✔
Price per user		$45 BUY	$75 BUY	~~$125~~ $85* BUY
10-user Team License		$250 BUY	$450 BUY	~~$750~~ $490* BUY
Enterprise License (unlimited users, up to 10 locations)		$450 BUY	$750 BUY	~~$1250~~ $850* BUY

LinqPad 유료 기능 사세요

- **무료 버전으로도 사용하는데 무리 없음**

- 자동 완성이 지원되면
  인생이 편해지고 세상살이가 유쾌해짐

- Line-by-Line debugging을 하게 되면
  스트레스를 적게 받아서 몸과 마음이 건강해짐

정리 및 결론

정리

- LinqPad가 왜 필요한지 설명
- LinqPad가 어디가 좋은지 설명
- LinqPad를 어떻게 쓰고 있는지 설명
- LinqPad를 더 잘 쓰기 위한 방법 소개

정리

- **LinqPad가 왜 필요한지 설명**
  - Live service에서 c# 스크립팅의 필요성
  - 기존 코드 참조가 필요하지만
    기존 프로젝트는 너무 무거움

정리

- LinqPad가 왜 필요한지 설명
- **LinqPad가 어디가 좋은지 설명**
  - 가볍고 빠르고 강력함
  - 기존 프로젝트와 호환이 가능하면서 독립적으로 존재

정리

- LinqPad가 왜 필요한지 설명
- LinqPad가 어디가 좋은지 설명
- **LinqPad를 어떻게 쓰고 있는지 설명**
  - 서비스 프로젝트 제외하고
    운영 과정에 필요한 프로그래밍은
    거의 다 linqpad로 하고 있음

정리

- LinqPad가 왜 필요한지 설명
- LinqPad가 어디가 좋은지 설명
- LinqPad를 어떻게 쓰고 있는지 설명
- **LinqPad를 더 잘 쓰기 위한 방법 소개**
  - 부가 기능과 plugin 들로 더 강력하게
  - 유료 버전까지 달린다면 금상첨화

결론

- **C# 을 쓴다면 LinqPad 꼭 쓰세요 !**

- 기존 프로젝트 dll을 참조해서 쓰면 매우 강력함

- 유료가 안 내킨다면, 무료 버전만으로도 충분함

부록. Live-service script 작성

Live-service script 작성 주의점 1

- 반드시,
  테스트 환경에서 동작을 확인하고
  실행 할 것.

- **변경 전** 테이블 등 **백업**은 필수! 제발 ㅠ

Live-service script 작성 주의점 2

- 스크립트가 **멈추는 경우에** 대한 고려 필수
  - 돌리고 보니 로직 오류가 있거나
  - 너어어어무 오래 걸린다거나
  - 실행 중 일부 데이터가 이상해서 Null Exception!

Live-service script 작성 주의점 3

- **잘못된 부분부터 다시 실행할 수 있는 구조** 고려
- 변경 및 적용 로그가 필요 (파일 출력 good)
- 지급 대상 유저 추출과 지급 단계를 분리
- 스크립트 실행 상태 기록(file) & 출력(console)
  - 유저 목록 및 필요 작업 기록
  - 변경 전/후 데이터 기록
  - 유저 데이터 업데이트 성공 여부 기록

Links

- LinqPad https://www.linqpad.net/
- Csscript http://www.csscript.net/

끝

lee@mobilfactory.co.kr

MOBIL FACTORY