



Entity Framework Core mit PostgreSQL

Speaker: Olaf Lischke

Speaker: Olaf Lischke



macht .NET, seit es .NET gibt



versucht, Projekte und Seminare zu kombinieren



singt Tenor in zwei Chören



zockte schon auf dem ZX 81, heute ausschließlich auf PC



fotografiert, seit er eine Kamera halten kann



fliegt, wenn Wetter und Zeit es zulassen (TMG/SEP)

Was? Postgres? Mal gehört, aber...

- Open Source Project
 - Seit 1986 University of California, Berkley
 - Erste öffentliche Release: 1989
- Einsatz und Distribution kostenlos
- Lange Lebensdauer der Versionen
 - Aktuelle Versionen: 10.22, 11.17, 12.12, 13.8, 14.5
 - Neue Versionen für 10, 11, 12, 13 **und** 14 ca. alle 3 Monate

8th September 2022: PostgreSQL 15 Beta 4 Released!

Wer benutzt denn sowas...?

- [StackOverflow Survey](#)
- **Unternehmen und Behörden**
 - Netflix
 - Uber
 - Rotes Kreuz (USA)...
- **Cloud-Anbieter**
 - Microsoft Azure
 - Google Cloud
 - Amazon



Was kann Postgres, was SQL-Server/Oracle nicht kann?

- Database Templating
- Table-Inheritance
- Einfache(?) .conf-Files
- SSL für die Client-Server-Kommunikation
- Vielfältige Replikationstechniken (a/synchron, gestreamt, logisch, physisch...)
- Multi-Version Concurrency Control (MVCC)
- Serializable Snapshot Isolation (SSI)
- Spricht auch Tcl, Perl, Java, Lua, sh, JavaScript
- Läuft auch auf Linux, Unix, Mac, BSD, Solaris...

Ich kann SQL – reicht das...?

- Ja. PostgreSQL benutzt zwar PL/pgSQL, aber die grundsätzliche Syntax ist sehr ähnlich zu T-SQL
- Teils andere Nomenklatur
- Tabellennamen in "" statt []
- Case Sensitive
- Schema-Angabe!

SQL Server	PostgreSQL
bit	Boolean
[var]char(n)	Text, bpchar, [var]char(n)
uniqueidentifier	ByteA
image	ByteA
rowversion	ByteA
binary	ByteA
geometry	Point, Line, LSeg, Box, Path, Polygon, Circle
?	json, jsonb

Zeitzone in PostgreSQL

- "UTC everywhere"

- Zeitangabe gilt immer als UTC-Zeitangabe, wenn nichts anderes angegeben.
- **Daher:** DateTime (C#) → timestampz, keine implizite Konvertierungen mehr.
- **Oder:** Selbst konfigurieren, ggf. DateTimeKind verwenden
- **Oder:** Altes Verhalten erzwingen:
`AppContext.SetSwitch("Npgsql.EnableLegacyTimestampBehavior", true);`

- DateOnly/TimeOnly-Unterstützung

- NodaTime-Unterstützung (<https://nodatime.org/>)

Bisher:

DateTime - Kind	PostgreSQL Typ
UTC	timestampz
Local	timestamp
Unspecified	timestamp

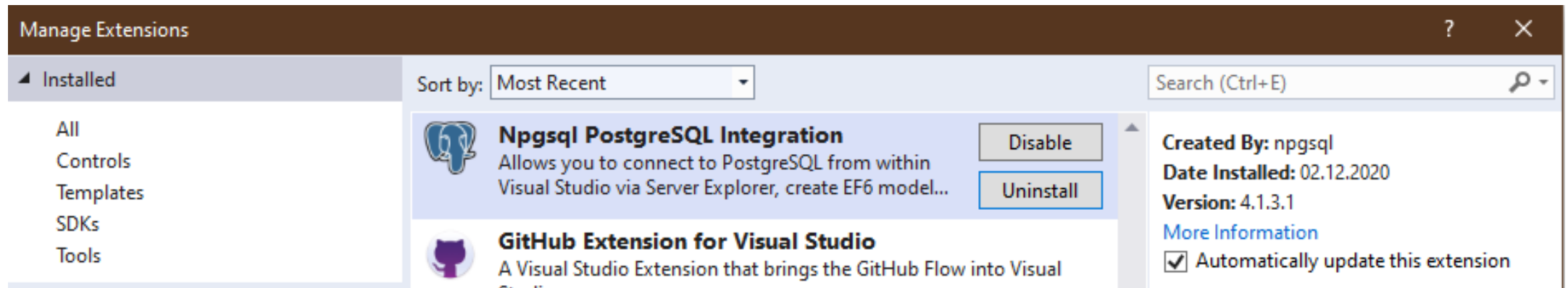
Gibt's ein Management Studio?

- Im Lieferumfang enthalten ist
 - pgAdmin
 - Browserbasiert
 - benutzt Admin Server (Erststart mit Geduld)
 - SQL Shell (psql)
 - Konsole
- Aber folgende können auch PostgreSQL:
 - Oracle SQL Developer
 - DBeaver (<https://dbeaver.io/>)
 - OmniDB (<https://omnidb.org/>)
 - Sqlectron (<https://sqlectron.github.io/>)

Open Source | | kostenlos

Kann Visual Studio mit Postgres...?

- ~~SQL Server Object Explorer~~ (weil kein SQL Server...)
- Server Explorer
 - Npgsql Postgres Extension <https://marketplace.visualstudio.com>



Und für EF Core gibt es einen Treiber?

- <https://github.com/npgsql> - Shay Rojansky (<https://github.com/roji>)
- Packages
 - `Npgsql.EntityFrameworkCore.PostgreSQL`
 - `Microsoft.EntityFrameworkCore.Tools`
 - `Microsoft.EntityFrameworkCore.Design`



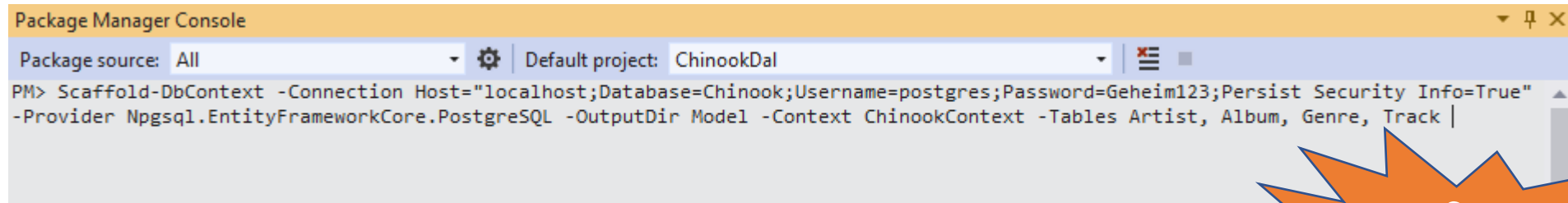
Was kann der Npgsql-Treiber? – Teil 1

- Generell

- `optionsBuilder.UseNpgsql(connectionString,`
 `opt =>`
 `{`
 `opt.EnableRetryOnFailure();`
 `opt.RemoteCertificateValidationCallback(...);`
 `opt.ProvideClientCertificatesCallback(...);`
 `});`

Was kann der Npgsql-Treiber? – Teil 2

- Reverse Engineering
 - Scaffold-DbContext **-Provider Npgsql.EntityFrameworkCore.PostgreSQL**



```
Package Manager Console
Package source: All Default project: ChinookDal
PM> Scaffold-DbContext -Connection Host="localhost;Database=Chinook;Username=postgres;Password=Geheim123;Persist Security Info=True"
-Provider Npgsql.EntityFrameworkCore.PostgreSQL -OutputDir Model -Context ChinookContext -Tables Artist, Album, Genre, Track |
```



- Code First - Bereitstellung
(aus Microsoft.EntityFrameworkCore.Infrastructure)
 - EnsureCreated()
 - Migrate()

Was kann der Npgsql-Treiber? – Teil 3

- Infrastruktur:
 - `modelBuilder.Entity<T>.UseXminAsConcurrencyToken();`
 - `modelBuilder.Entity<T>.HasComment(„Kommentar“);`

aus: Microsoft.EntityFrameworkCore

Was kann der Npgsql-Treiber? – Teil 4

aus: Microsoft.EntityFrameworkCore

- EntityTypeConfiguration
 - FluentApi Konfiguration (OnModelCreating) je Typ in eigene, übersichtliche Datei auslagern:

1 Verweis | Olaf Lischke, vor 7 Tagen | 1 Autor, 1 Änderung

```
internal class EmployeeConfiguration : IEntityTypeConfiguration<Employee>
{
```

0 Verweise | Olaf Lischke, vor 7 Tagen | 1 Autor, 1 Änderung

```
public void Configure(EntityTypeBuilder<Employee> builder)
{
```

```
    builder.Property(e => e.EmployeeId).ValueGeneratedNever();
```

```
    builder.HasOne(d
```

```
        .WithMany(p =
```

```
        .HasForeignKey
```

```
        .HasConstraint
```

in OnModelCreating:

```
// EntityTypeConfigurations
```

```
modelBuilder.ApplyConfiguration(new EmployeeConfiguration());
```

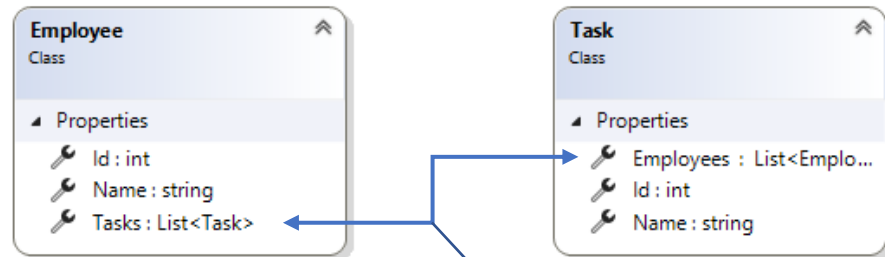
```
modelBuilder.ApplyConfiguration(new EmployeeJsonConfiguration());
```

```
modelBuilder.ApplyConfiguration(new OrderConfiguration());
```

Was kann der Npgsql-Treiber? - Teil 5

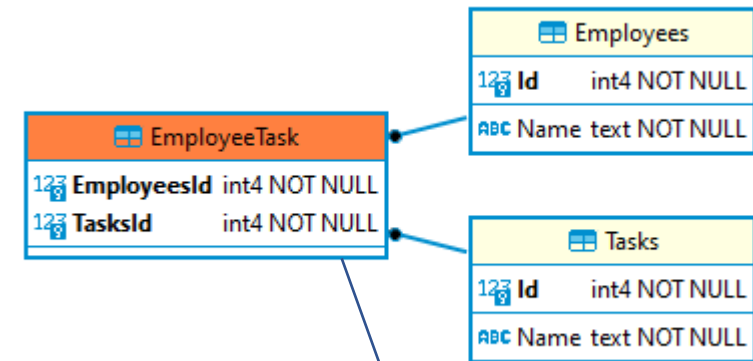
- N:M-Zuordnungen

Objektmodell



Ein Mitarbeiter hat viele Tasks,
an einem Task arbeiten viele Mitarbeiter

Datenbankschema



Enthält das Modell nur Zuordnung,
wird Zuordnungstabelle angelegt.

Auch Reverse
Engineered!

Was kann der Npgsql-Treiber? – Teil 6

- JSON-Datentypen:
 - json – großes Textfeld/BLOB
 - jsonb
 - JSON-Objekte
 - inkl. Indizierung und [Query-Syntax](#)

123 employee_id	JSON data
1	{"LastName":"Davolio","FirstName":"Nancy","Title":"Sales Representati
2	{"LastName":"Fuller","FirstName":"Andrew","Title":"Vice President, Sale
3	{"LastName":"Leverling","FirstName":"Janet","Title":"Sales Representati
4	{"LastName":"Peacock","FirstName":"Michael","Title":"Sales Represent
5	{"LastName":"Buchanan","FirstName":"Margaret","Title":"Sales Manager"
6	{"LastName":"Suyari","FirstName":"David","Title":"Sales Representative"
7	{"LastName":"King","FirstName":"Adrian","Title":"Sales Representative"
8	{"LastName":"Callahan","FirstName":"Heather","Title":"Sales Coordinator"
9	{"LastName":"Dodds","FirstName":"Jennifer","Title":"Sales Representative"

EmployeesJson
Class
Properties

- Data : string
- EmployeeId : short

```
EmployeesJson nancy = context.EmployeesJsons
    .FirstOrDefault(
        emp => EF.Functions.JsonContains(emp.Data, "{\"FirstName\":\"Nancy\"}")
    );
```

```
EmployeesJson andrew = context.EmployeesJsons
    .FromSqlRaw("SELECT * FROM employeesjson WHERE data ->> 'FirstName' = 'Andrew'")
    .FirstOrDefault();
```

Ausblick auf EF Core 7

- Beschleunigte Massenupdates

- ExecuteUpdate
- ExecuteDelete



- Json-Columns für alle

- Json-Columns-Unterstützung in LINQ → Änderungen am Provider
- NICHT gleich Npgsql-Json-Columns aus EF Core 6!

Vielen Dank!

Slides auf

<https://github.com/olaflischke/basta2022>

Für spätere Fragen:

olaf.lischke@lischke-edv.de