Backend Database Documentation

Table of Contents

[Database 1](#_Toc121916128)

[Overview 1](#_Toc121916129)

[Migrations 2](#_Toc121916130)

[Building the Database 2](#_Toc121916131)

[Seed Data 2](#_Toc121916132)

[BrickLink XML Files 2](#_Toc121916133)

[Load Seed Data 3](#_Toc121916134)

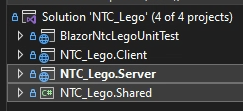
[Load Seed Data (Remote) 3](#_Toc121916135)

[Getting Data from the Backend to the Frontend 4](#_Toc121916136)

# Database

## Overview

The database for the NTC\_Lego solution is built using Entity Framework migrations and populated with seed data using a SQL script. The DataContext and DataServices classes are located within the NTC\_Lego.Server project, while the Models and ViewModels are inside the NTC\_Lego.Shared project.



Model classes are created to act as placeholders for database objects. The DataContext class is used to define these Models into tables and to specify foreign key relations and constraints. Seed data may also be defined in this class; however, this project does not use this feature because our seed data is to too large to fit within a migration.

The DataService classes are used to perform basic CRUD operations on the database through the DataContext. This class may also be used to map database Models into ViewModels to avoid circular dependencies due to nested foreign key relations. Mapping is performed within the DataContext calls, using the Linq Select statement.

## Migrations

If you have made new Models and/or changed the DataContext, a new database migration is needed for these changes to be applied to the database.

To create a new migration, you must open a terminal in the NTC\_Lego.Server project and run the following command:

**dotnet ef migrations add migration-name**

One that command has successfully run, you will need to apply that change to the database by running the command:

**dotnet ef database update**

If you have made major changes to the Models and DataContext, you may need to empty the Migrations folder within the server project, and then create a new initial migration.

## Building the Database

If you are rebuilding the database after already creating it once, ensure you have deleted your old database before following these instructions.

1. Open the project solution in Visual Studio
2. Open a terminal in the NTC\_Lego.Server project and run the command:
   1. **dotnet ef database update**
3. Open SQL Server Management Studio (SSMS) and locate your new database labeled:
   1. **ntcLego-Dev**
4. Success, you now have the development database on your localhost. In the next section we will go over how to populate the database with seed data.

# Seed Data

The majority of the seed data used to populate the database is contained within XML files taken from the BrickLink (<https://www.bricklink.com/>) website. Before attempting to load the seed data into the database, ensure you have the most up-to-date XML files.

## BrickLink XML Files

To retrieve the most recent BrickLink XML files follow these instructions:

1. Login to BrickLink
2. Go to <https://www.bricklink.com/siteMap.asp> and click “Download”
   1. Graphical user interface

      Description automatically generated with medium confidence
3. Download Catalog Items (Sets, Parts), Item Types, Categories, and Colors
   1. Graphical user interface

      Description automatically generated
4. Replace the old XMLs in the repo at “**repoName\Documents\BrickLink\XML**” with these new files.

## Load Seed Data

After you have created the database using Entity Framework migrations, you will need to populate it with seed data using the following instructions:

1. Open SSMS and create a new query in the ntcLego-Dev database
2. Locate the script file in the repo at the following location:
   1. **repoName\Documents\BrickLink\loadXMLSeedData.sql**
3. Open this script within the ntcLego-Dev database. Follow along with the instructions written at the top of the script file, most importantly make sure to placer the XML files at the location “C:\temp\” on your machine
4. Execute the command; it should complete within 10-15 seconds and show confirmation that multiple rows have been changes.

## Load Seed Data (Remote)

To populate a remote database with seed data, you must use the following file:

**repoName\Documents\BrickLink\loadXMLSeedData(REMOTE).sql**

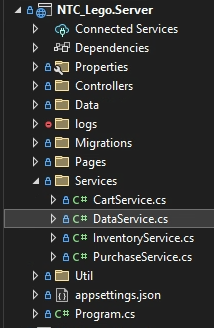
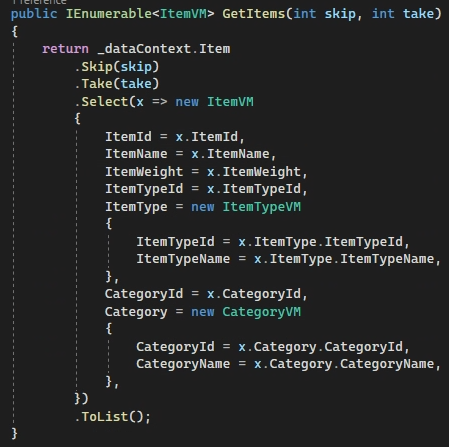
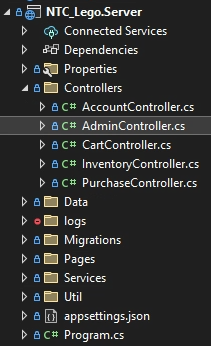
The largest difference in this process is that with a remote database, you cannot upload locally stored file data directly. You must host the XMLs in a remote storage container and then link to the external data source.

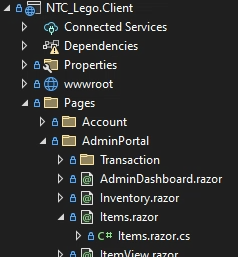
More detailed instructions are included at the top of the script file.

# Getting Data from the Backend to the Frontend

The process of retrieving data from the backend and placing it on the Blazor frontend requires multiple steps. The following instructions detail this process:

1. Create the necessary Models and/or ViewModels, perform an Entity Framework migration if any database changes have occurred
   1. Text

      Description automatically generated
2. Create a DataService method which performs basic CRUD operation on the DataContext
   1.  
3. Create a Controller method which uses the DataService method and specify an endpoint
   1. Business logic and validation should be performed within the controller
   2.  Text

      Description automatically generated
4. Use an HttpClient on the Blazor page to connect to the endpoint, display the data on the page
   1.  Text

      Description automatically generated
   2. 