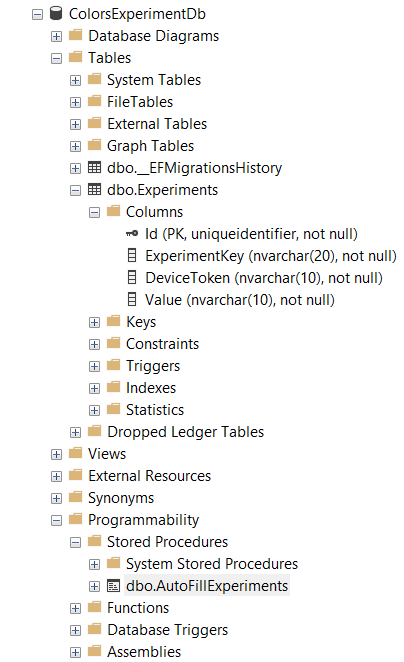
Database structure:  
Database consists of 2 tables Experiments and EFMigrations(because at first I used code-first approach)   


Experiments table built with 4 fields such as:  
1. Id primary key

2. ExperimentKey name of experiment

3. DeviceToken unique client Identifier

4. Value corresponding experimental value

Also database has stored procedure to immitate load with a large amount of data:  
PROCEDURE [dbo].[AutoFillExperiments]

AS

BEGIN

DECLARE @Counter INT = 1;

-- Вставка 5000 записів для експерименту з кнопкою кольору

WHILE @Counter <= 5000

BEGIN

DECLARE @ButtonColor NVARCHAR(7);

IF RAND() \* 100 < 33.3

SET @ButtonColor = '#FF0000';

ELSE IF RAND() \* 100 < 66.6

SET @ButtonColor = '#00FF00';

ELSE

SET @ButtonColor = '#0000FF';

INSERT INTO Experiments (Id, ExperimentKey, DeviceToken, Value)

VALUES (NEWID(), 'button\_color', CAST(@Counter AS NVARCHAR(20)), @ButtonColor);

SET @Counter = @Counter + 1;

END;

-- Вставка 5000 записів для експерименту з ціною

WHILE @Counter <= 10000

BEGIN

DECLARE @Price DECIMAL(10, 2);

DECLARE @PriceSelector DECIMAL(10, 2) = RAND() \* 100;

IF @PriceSelector < 75

SET @Price = 10;

ELSE IF @PriceSelector < 85

SET @Price = 20;

ELSE IF @PriceSelector < 90

SET @Price = 50;

ELSE

SET @Price = 5;

INSERT INTO Experiments (Id, ExperimentKey, DeviceToken, Value)

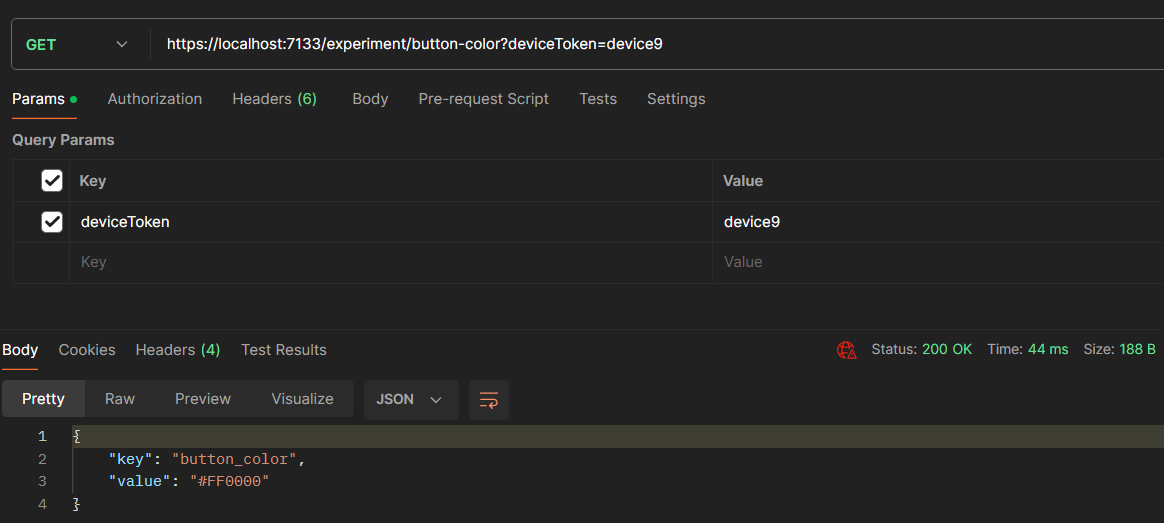
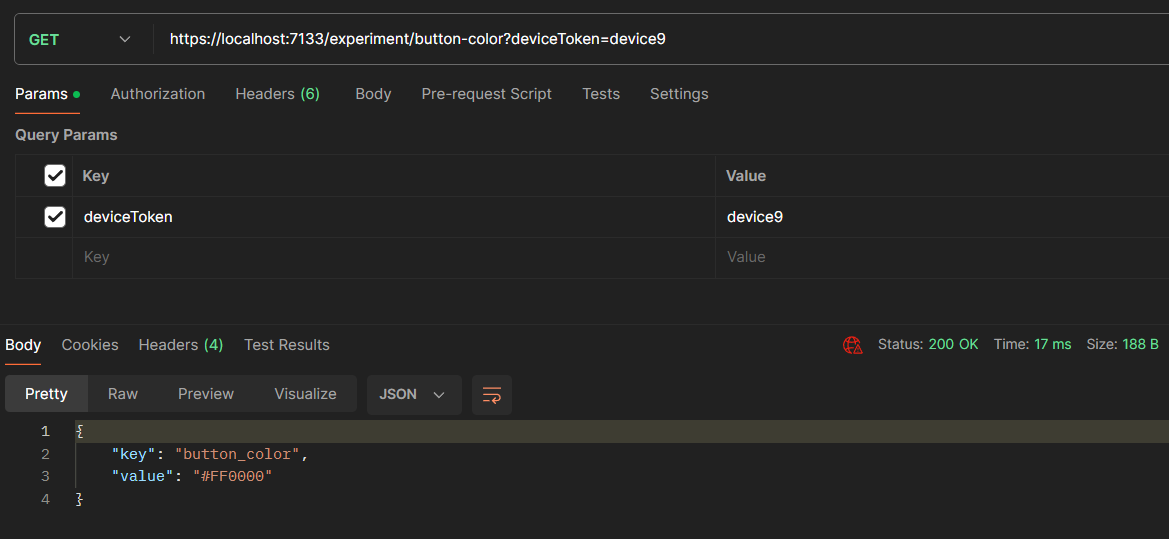
VALUES (NEWID(), 'price', CAST(@Counter AS NVARCHAR(20)), CAST(@Price AS NVARCHAR(20)));

SET @Counter = @Counter + 1;

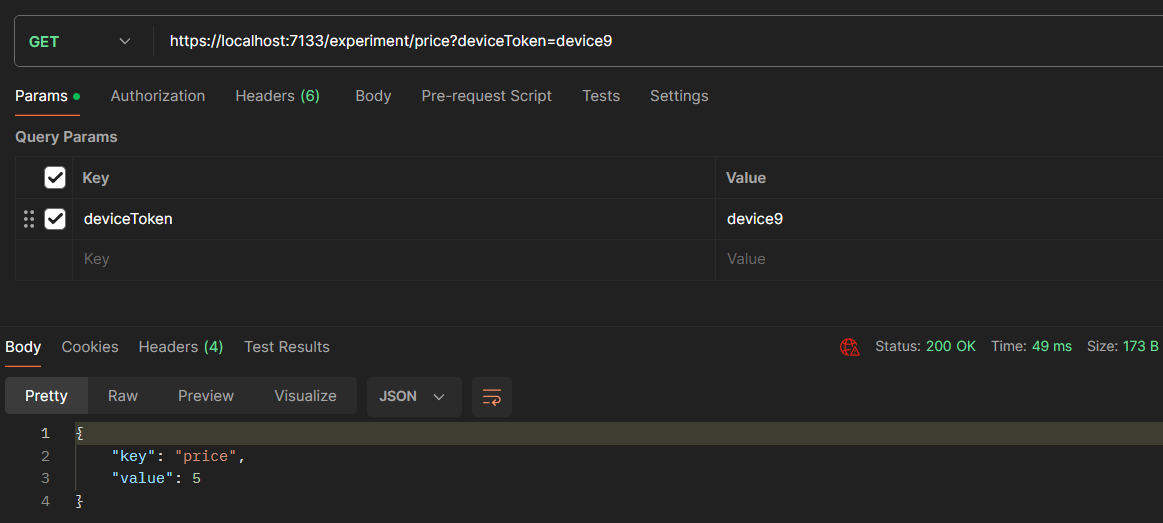
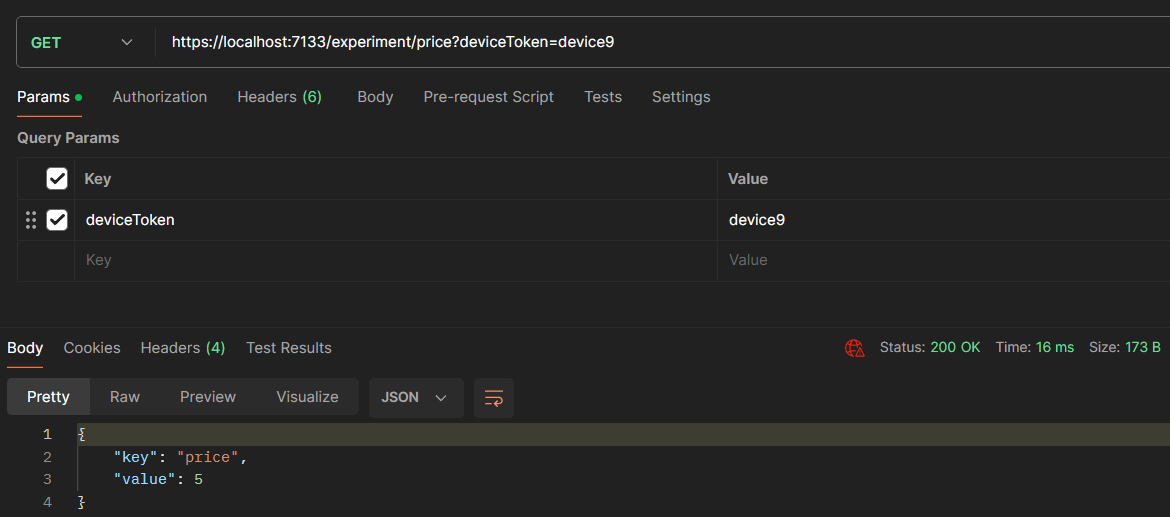
END;

END;

Debugging endpoints from postman:  
Button experiment:

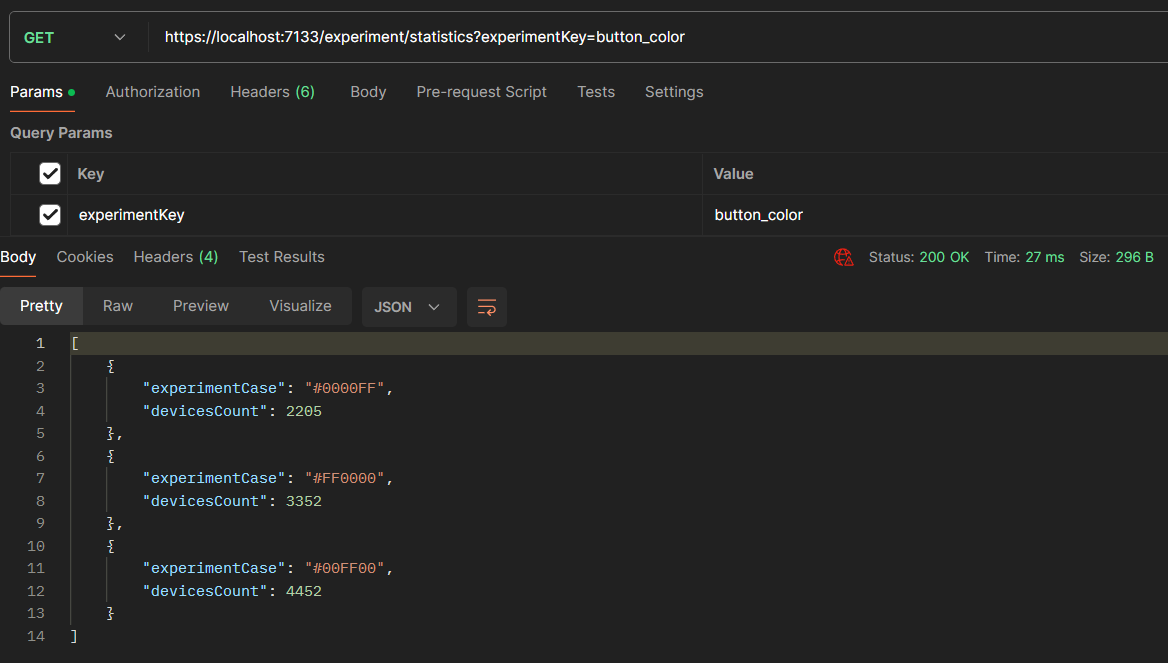
* new device:  
  
* db has same device:  
  

Price experiment:

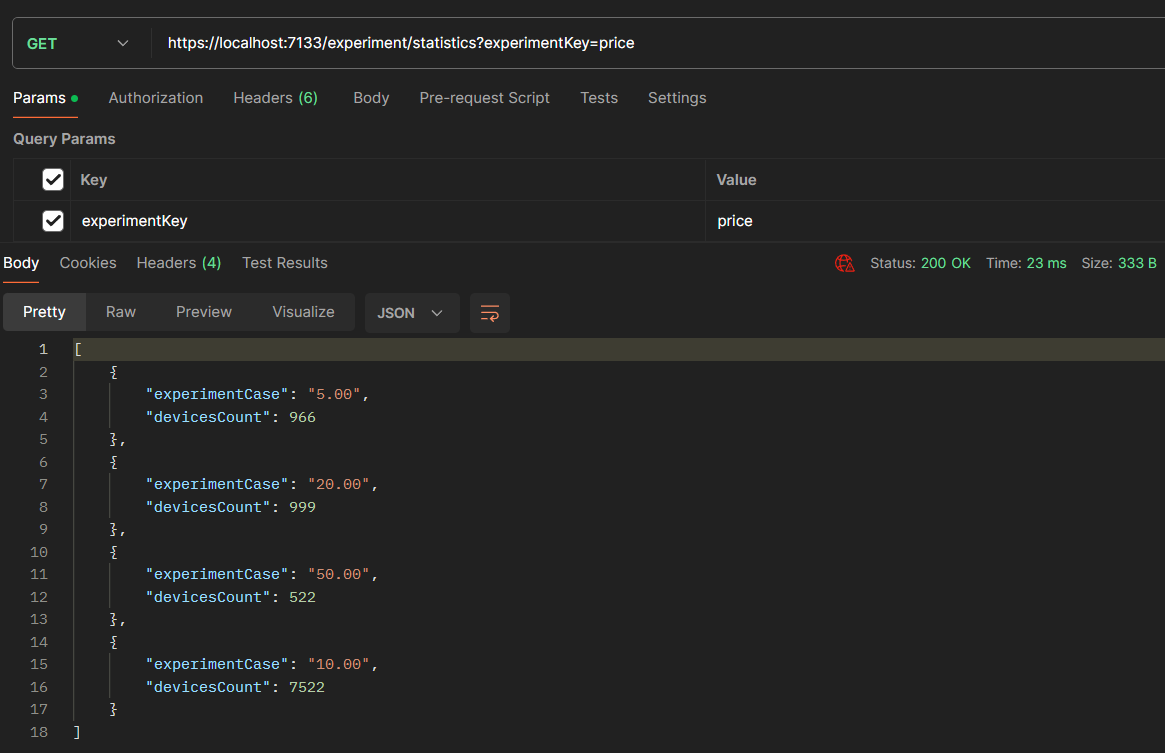
* new device:  
  
* db has same device:  
  

Experiments stats:

* Button color experiment



* Price experiment



Benchmark results:

