

СОДЕРЖАНИЕ

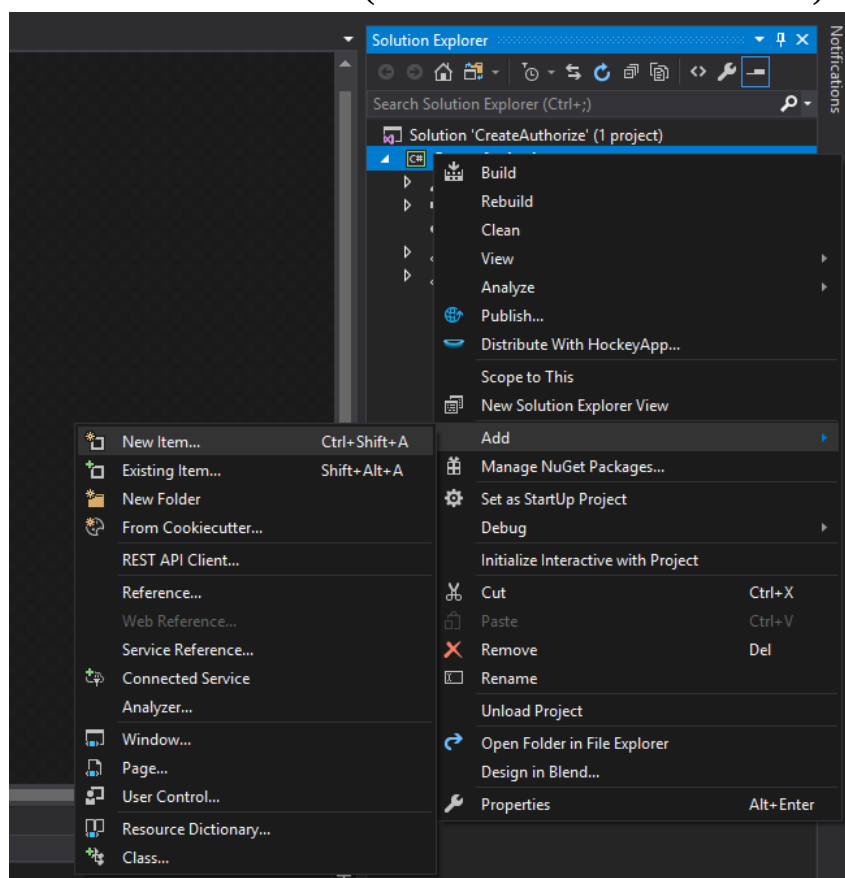
1 Разработка Базы данных	3
1.1 Создание таблиц	3
2 Создание приложения	12
2.1 Создание окна авторизации	12
2.2 Создание окна регистрации	16
2.3 Добавление стилей	18

1 Разработка Базы данных

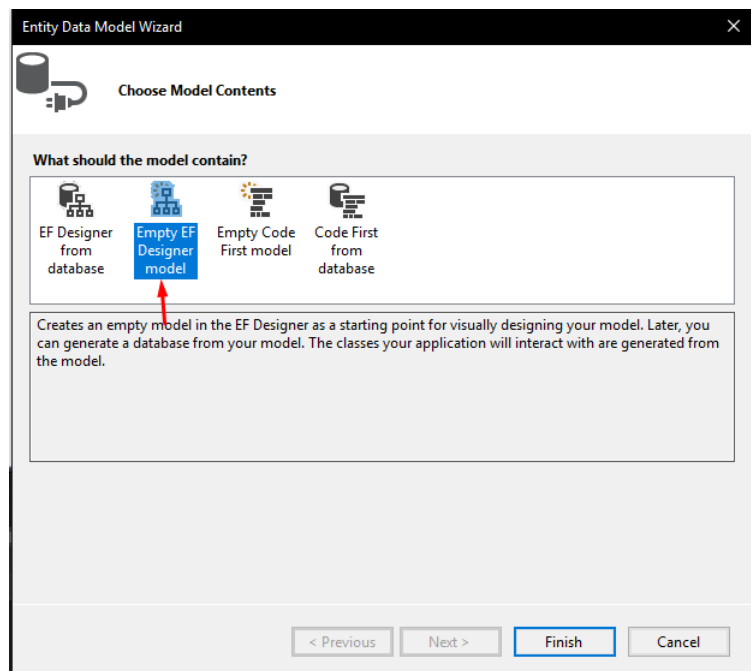
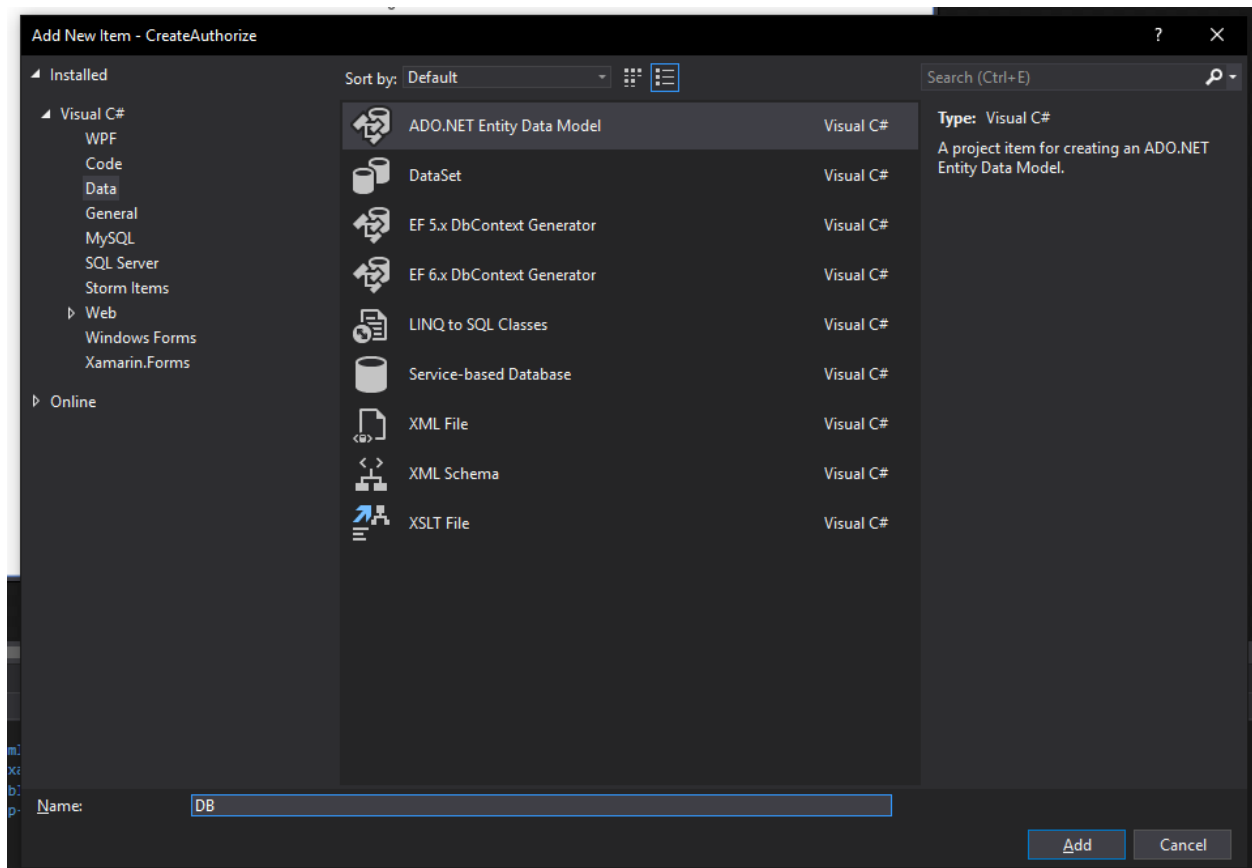
1.1 Создание таблиц

Создадим приложение WPF назовём его «CreateAuthorize».

Теперь добавим модель ADO.NET. Для этого Правой кнопкой мыши по проекту → Add → New item... (или сочетание Ctrl+Shift+A).

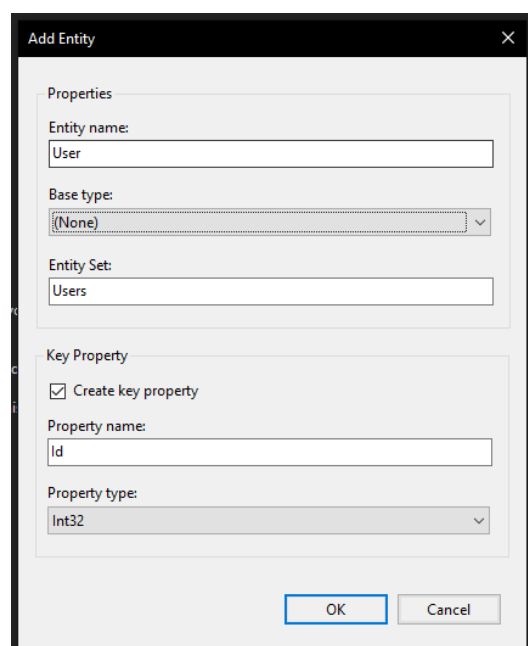
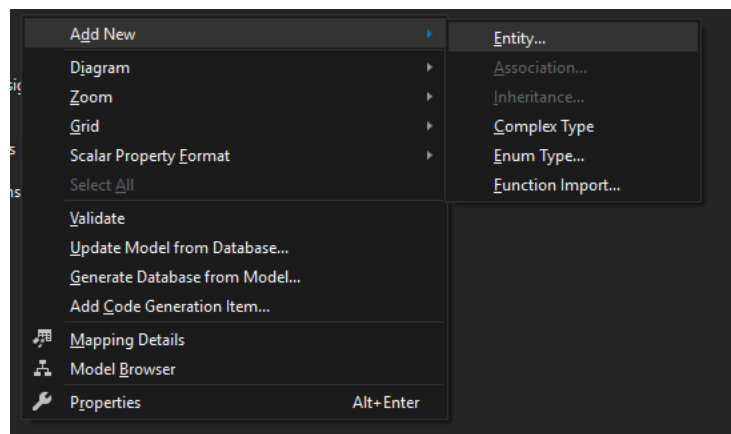


После появления нового окна → выбираем Data → ADO.NET Entity Data Model в качестве имени назначим «DB» → жмём Add.



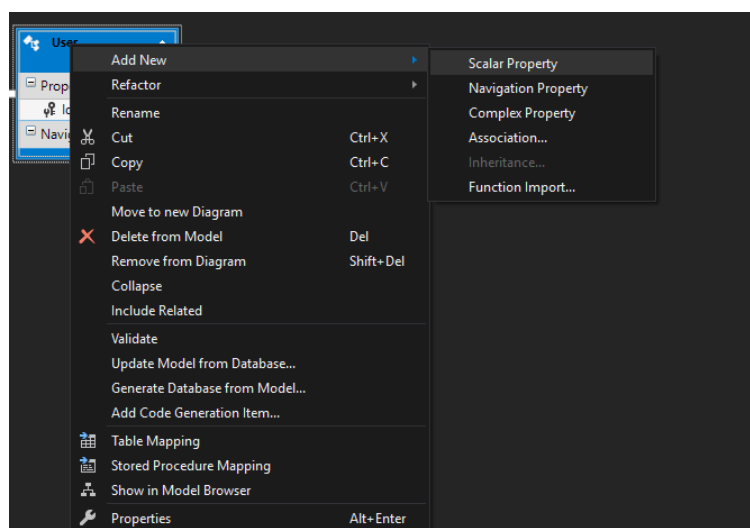
Выберем Empty EF Designer model → жмём Finish

После появления добавим новую таблицу: Правой кнопкой мыши (ПКМ) → Add New → Entity

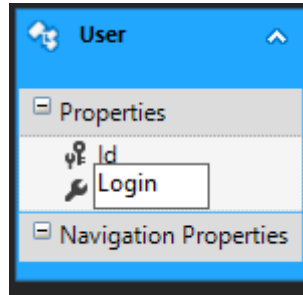


Назовём нашу таблицу → жмём ОК.

Теперь добавим новые поля в нашу таблицу: ПКМ по таблице → Add New → Scalar Property.

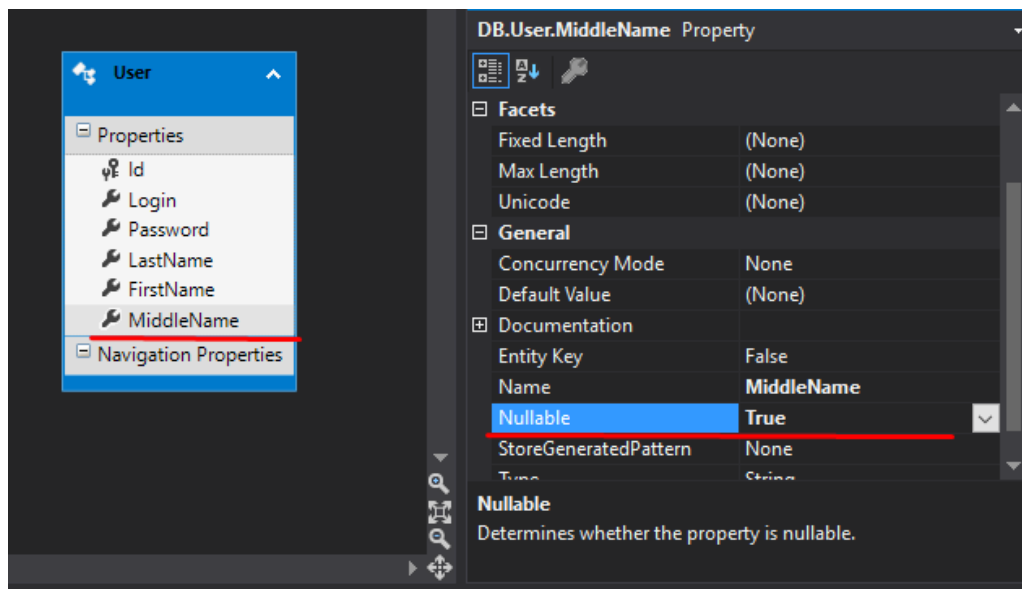


Присвоим имя новому полю «Login» → жмём Enter.

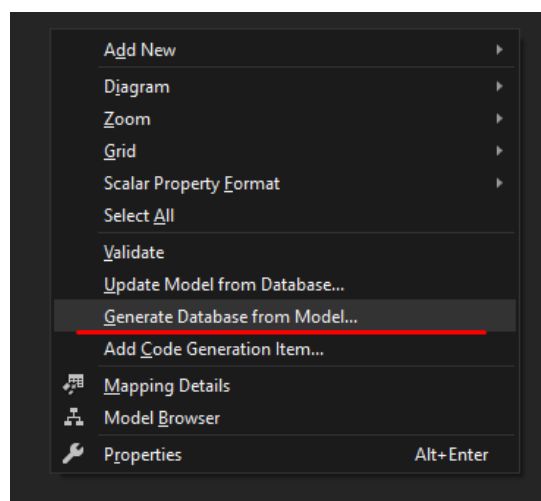


Теперь добавим такие поля как «Password», «LastName», «FirstName», «MiddleName».

Изменим свойство обязательности у поля «MiddleName», чтобы разрешено было хранить NULL. Выделив поле переходим в его свойства и меняем свойство «Nullable» на True.

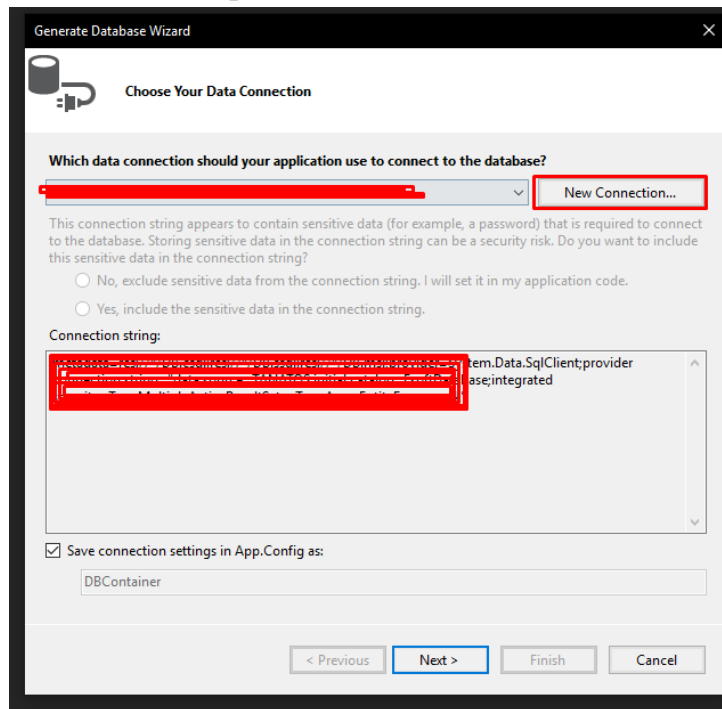


Теперь создадим нашу базу данных назовём её «TestAuthorize». Для этого



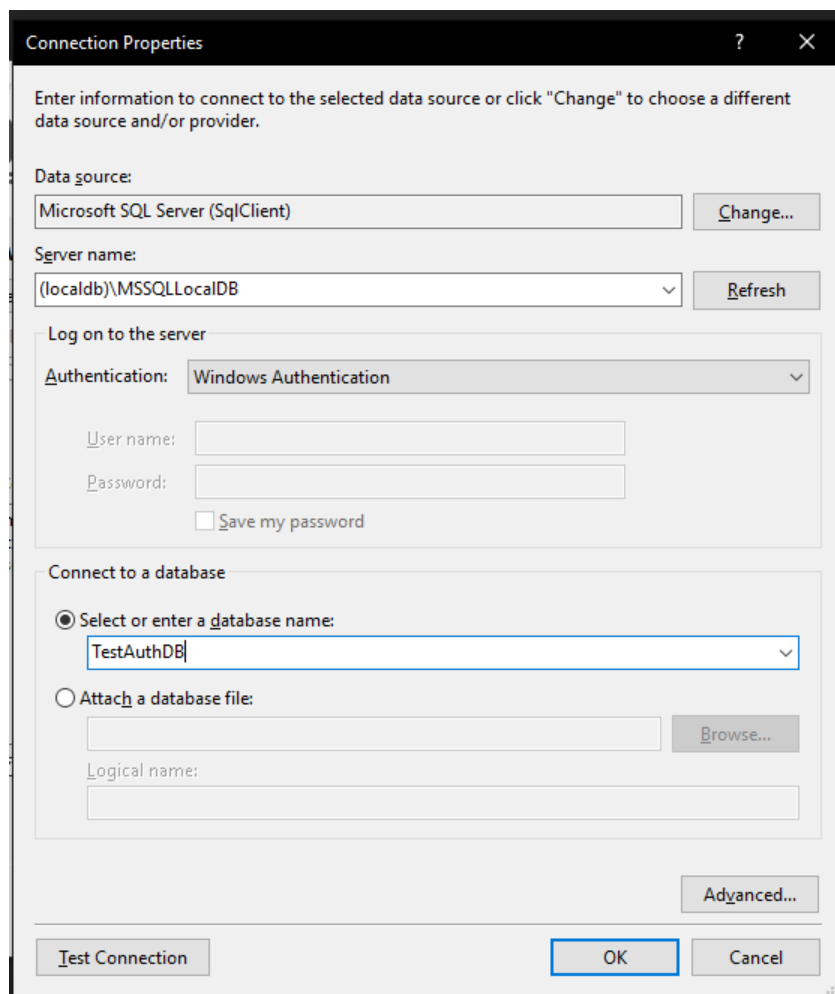
ПКМ по пустой рабочей области и выберем Generate Database from Model...

В открывшемся окне выбираем New Connection.



В поле «Server name:» указываем «(localdb)\MSSQLLocalDB», или свой сервер БД

В имени базы данных пишем «TestAuthDB» и жмём ОК

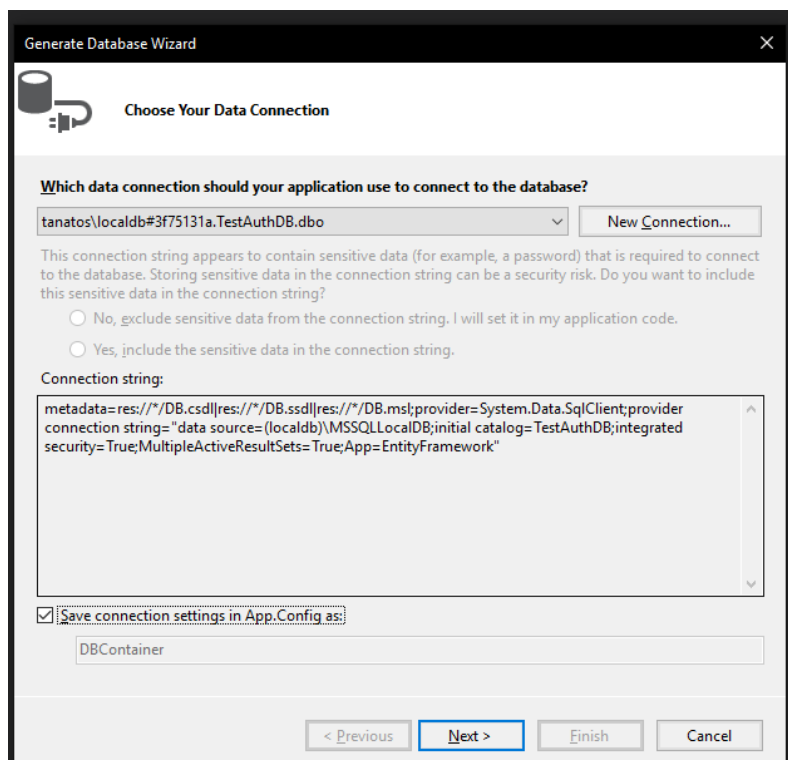


Появится окошко с вопросом, нужно ли создавать БД, отвечаем «Yes»

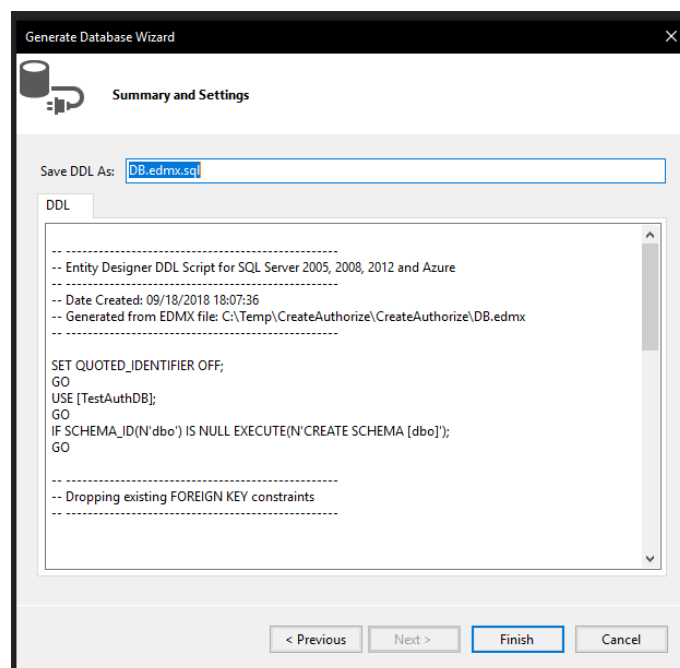
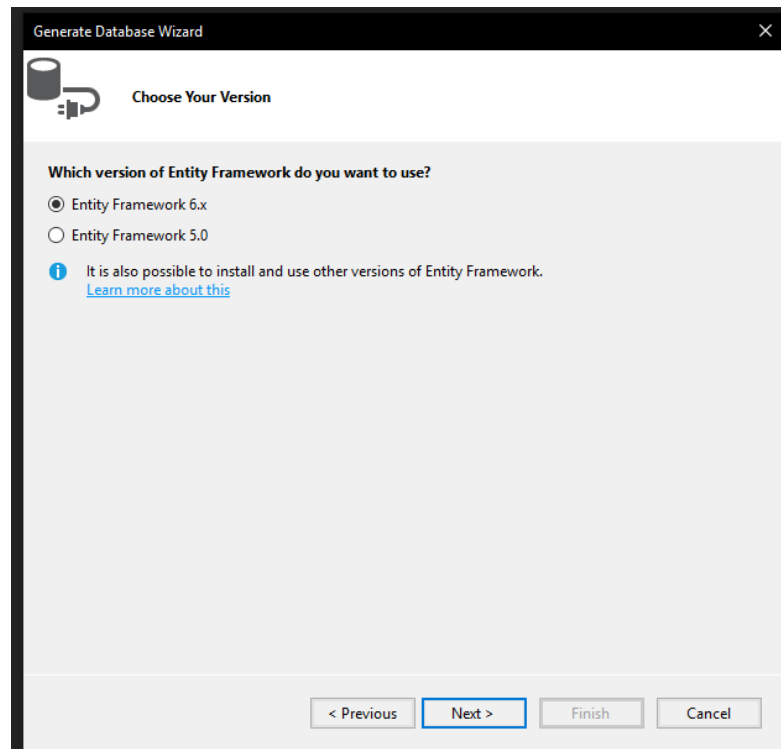


После некоторого времени ожидания наше соединение будет создано.

Далее просто жмём Next.

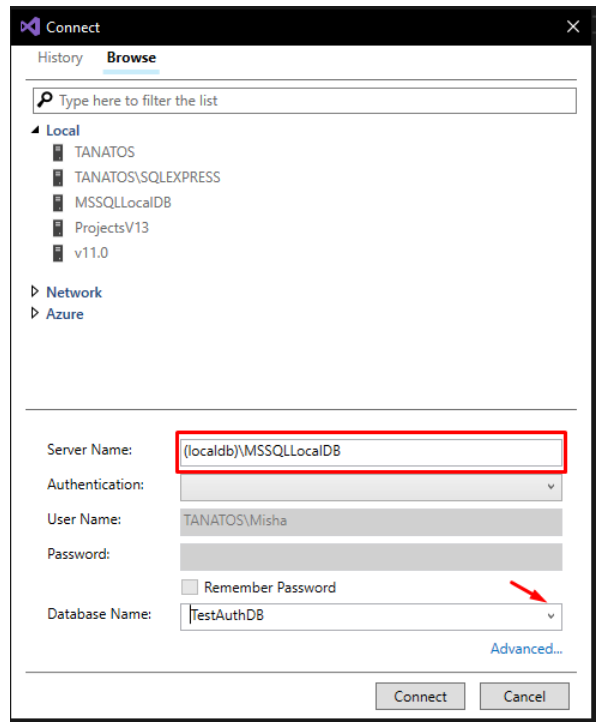
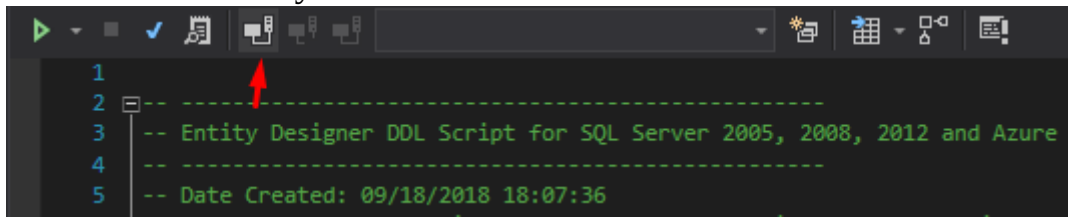


Далее жмём Next

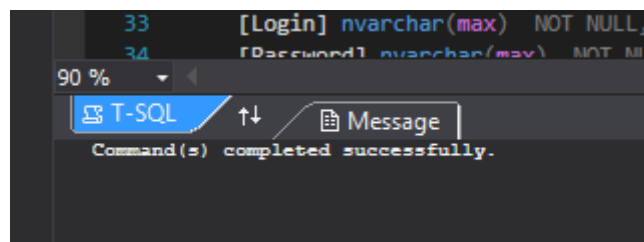


И жмём Finish.

Нажмём на кнопочку.



Выберем наш сервер БД, и выберем саму базу данных → жмём Connect.
После подключения к БД жмём зелёную стрелочку.



Когда появится сообщение что ниже, наша бд будет создана.

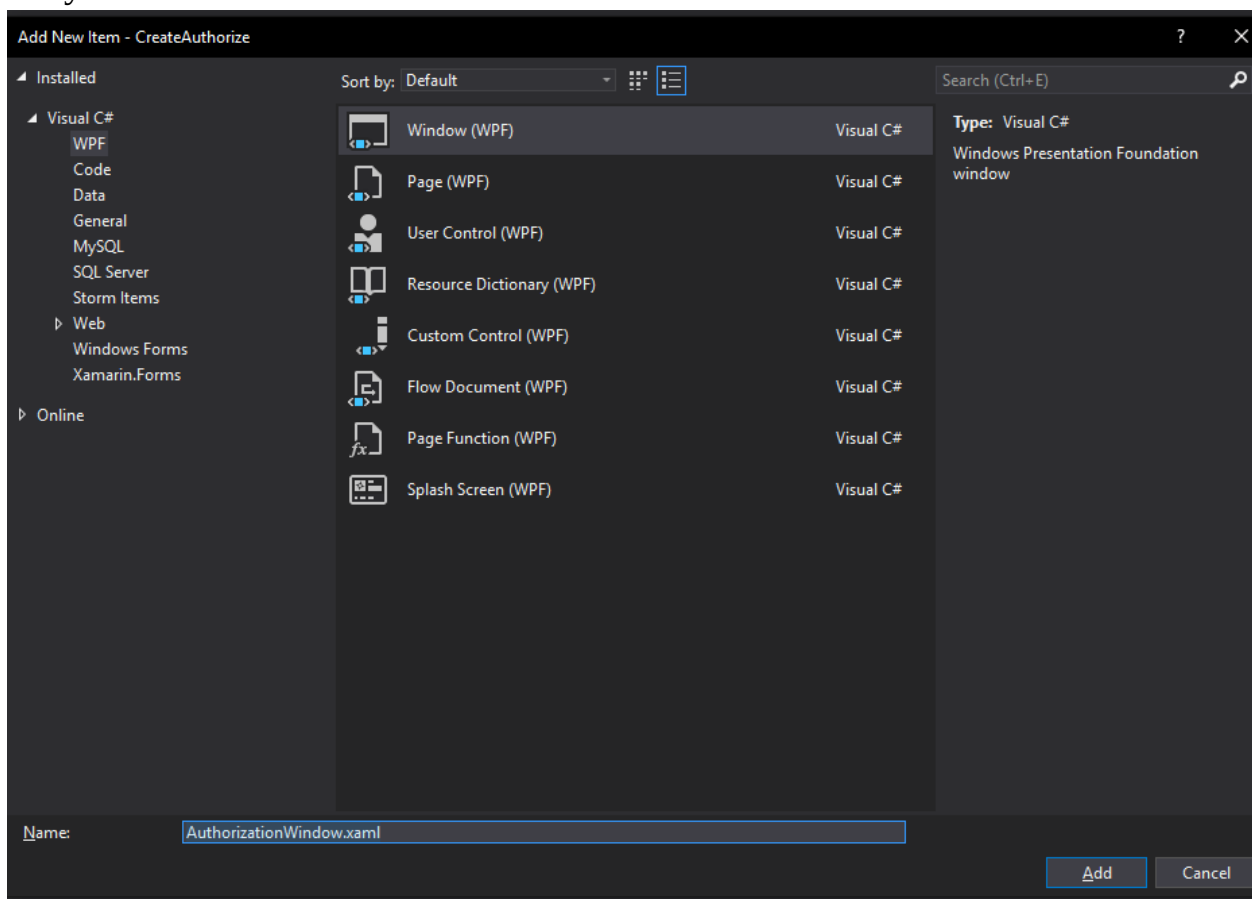
2 Создание приложения

2.1 Создание окна авторизации

В окне авторизации будет поле ввода Логина и пароля, а также 2 кнопки Войти и Зарегистрироваться. Из этого следует что нам нужно поделить форму на 3 строчки и 2 столбца.

Для этого жмём ПКМ по названию проекта → Add → New item или Ctrl + Shift + A.

В появившемся окне выбираем WPF → и Window (WPF) в качестве имени указываем «AuthorizationWindow.xaml» → жмём Add.



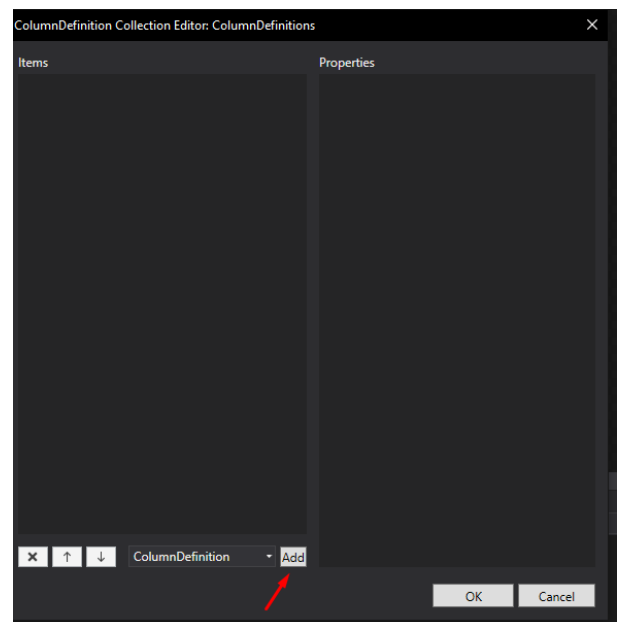
Зададим размеры нашему окну его высота будет 200, а ширина 400. Так же назначим стартовую позицию окна по центру экрана, а в качестве заголовка напишем «Авторизация».

Это код вёрстки программы на данный момент.

```
<Window x:Class="CreateAuthorize.AuthorizationWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CreateAuthorize"
        mc:Ignorable="d"
        Title="Авторизация" Height="200" Width="400" WindowStartupLocation="CenterScreen">
    <Grid>
    </Grid>
</Window>
```

Теперь добавим колонки и строки для этого ставим курсор на теге «Grid» и переходим на его свойства и там раскрываем список Layout. В нём находим свойство ColumnDefinitions и RowDefinitions, эти свойства отвечают за количество строк и колонок в Grid. Нажав на кнопку возле свойства ColumnDefinitions откроется окно, в нём вы можете менять количество столбцов по клику по кнопке Add. Теперь добавим 2 столбца и жмём ОК.

Тоже самое делаем со строками, т.е. со 2 свойством. Создаём 3 строки



После этого наш код будет выглядеть следующим образом:

```
<Window x:Class="CreateAuthorize.AuthorizationWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CreateAuthorize"
        mc:Ignorable="d"
        Title="Авторизация" Height="200" Width="400" WindowStartupLocation="CenterScreen">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition/>
            <RowDefinition/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>
    </Grid>
</Window>
```

Теперь добавим следующие элементы:

TextBlock – простой текст.

TextBox – поле для ввода текста.

PasswordBox – поле для ввода пароля.

Button – кнопка.

```
<Window x:Class="CreateAuthorize.AuthorizationWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CreateAuthorize"
        mc:Ignorable="d"
        Title="Авторизация" Height="200" Width="400" WindowStartupLocation="CenterScreen">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition/>
            <RowDefinition/>
            <RowDefinition/>
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition/>
            <ColumnDefinition/>
        </Grid.ColumnDefinitions>
        <TextBlock Text="Логин"/>
        <TextBlock Text="Пароль" Grid.Row="1"/>
        <TextBox Name="login" Grid.Column="1" Grid.Row="0"/>
        <PasswordBox Name="password" Grid.Column="1" Grid.Row="1"/>
        <Button Grid.Row="2" Grid.Column="0" Content="Войти" Click="AuthorizationClick"/>
        <Button Grid.Row="2" Grid.Column="1" Content="Зарегистрироваться" Click="RegistrationClick"/>
    </Grid>
</Window>
```

Здесь опишу свойства:

Text – свойство которое указывает отображаемый текст.

Grid.Row – указывает позицию элемента в Grid, т.е. номер строки где расположен элемент, отсчёт начинается с 0.

Grid.Column – указывает позицию элемента в Grid, т.е. номер столбца где расположен элемент, отсчёт начинается с 0.

Content – свойство позволяет указать внутри элемента отображаемый текст, так как у таких элементов отсутствует свойство Text.

Name – даёт имя элементу чтобы взаимодействовать с ним в коде C#.

Click – событие, внутри кавычек пишется названия события. Для его создания надо поставить на него курсор и нажать F12 если не работает F12, то ПКМ → Go to definitions (перейти к определению).

Теперь инициализируем события Click согласно описанию, данному выше.

В результате у нас появятся 2 события:

```
private void AuthorizationClick(object sender, RoutedEventArgs e)
{
}

namespace CreateAuthorize
{
    /// <summary>
    /// Interaction logic for AuthorizationWindow.xaml
    /// </summary>
    public partial class AuthorizationWindow : Window
    {
        DBContainer db;
        public AuthorizationWindow()
        {
            InitializeComponent();
            db = new DBContainer();
        }

        private void AuthorizationClick(object sender, RoutedEventArgs e)
        {
        }

        private void RegistrationClick(object sender, RoutedEventArgs e)
        {
        }
    }
}
```

Теперь подключим в открывшемся файлике нашу базу данных:

Добавим проверку на пустые поля и проверку пароля и логина в БД, при клике:

```
if (login.Text == "" || password.Password == "")
{
    MessageBox.Show("Ошибка пустые поля");
    return;
}
if(db.Users.Select(item => item.Login + " " + item.Password).Contains(login.Text + " " + password.Password))
{
    MessageBox.Show("Вы авторизованы");
}
else
{
    MessageBox.Show("Ошибка логина/пароля");
}
```

2.2 Создание окна регистрации

Его имя «RegistrationWindow.xaml».

Его свойства: высота=400, ширина=400, стартовая позиция так же по центру, заголовок «Регистрация».

Количество колонок – 2, а строк – 6.

Так же в первой колонке находятся TextBlock всего их 5:

Логин, Пароль, Фамилия, Имя, Отчество.

Во 2 колонке 5 TextBox их свойства Name:

login, lastName, firstName, middleName и один PasswordBox (Name=password).

И снизу 2 Button: «Зарегистрироваться» и «Отмена», их события Click: «RegistrationClick» и «CancelClick».

В результате у вас будет такой код:

```
<Window x:Class="CreateAuthorize.RegistrationWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  xmlns:local="clr-namespace:CreateAuthorize"
  mc:Ignorable="d"
  Title="Регистрация" Height="400" Width="400" WindowStartupLocation="CenterScreen">
  <Grid>
    <Grid.RowDefinitions>
      <RowDefinition/>
      <RowDefinition/>
      <RowDefinition/>
      <RowDefinition/>
      <RowDefinition/>
      <RowDefinition/>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
      <ColumnDefinition/>
      <ColumnDefinition/>
    </Grid.ColumnDefinitions>
    <TextBlock Text="Логин" Grid.Row="0" Grid.Column="0"/>
    <TextBlock Text="Пароль" Grid.Row="1" Grid.Column="0"/>
    <TextBlock Text="Фамилия" Grid.Row="2" Grid.Column="0"/>
    <TextBlock Text="Имя" Grid.Row="3" Grid.Column="0"/>
    <TextBlock Text="Отчество" Grid.Row="4" Grid.Column="0"/>
    <TextBox Name="login" Grid.Row="0" Grid.Column="1"/>
    <PasswordBox Name="password" Grid.Row="1" Grid.Column="1"/>
    <TextBox Name="lastName" Grid.Row="2" Grid.Column="1"/>
    <TextBox Name="firstName" Grid.Row="3" Grid.Column="1"/>
    <TextBox Name="middleName" Grid.Row="4" Grid.Column="1"/>
    <Button Content="Зарегистрироваться" Grid.Row="5" Grid.Column="0" Click="RegistrationClick"/>
    <Button Content="Отмена" Grid.Row="5" Grid.Column="1" Click="CancelClick"/>
  </Grid>
</Window>
```

Теперь напишем для него бизнес логику (код С#). Он будет содержать проверку на пустые поля, проверку на наличие логина в БД и процесс регистрации, а также оповещение об успешной регистрации. После завершения регистрации вернёмся на форму авторизации.

```
/// <summary>
/// Interaction logic for RegistrationWindow.xaml
/// </summary>
public partial class RegistrationWindow : Window
{
    DBContainer db = new DBContainer();
    public RegistrationWindow()
    {
        InitializeComponent();
    }

    private void RegistrationClick(object sender, RoutedEventArgs e)
    {
        if (login.Text == "" || password.Password == "" || lastName.Text == "" || firstName.Text == "" || middleName.Text == "")
        {
            MessageBox.Show("Пустые поля");
            return;
        }
        if (db.Users.Select(item => item.Login).Contains(login.Text))
        {
            MessageBox.Show("Такой логин существует в системе");
            return;
        }
        User newUser = new User() {
            Login = login.Text,
            Password = password.Password,
            LastName = lastName.Text,
            FirstName = firstName.Text,
            MiddleName = middleName.Text
        };
        db.Users.Add(newUser);
        db.SaveChanges();
        MessageBox.Show("Вы успешно зарегистрировались");
        AuthorizationWindow aw = new AuthorizationWindow();
        aw.Show();
        this.Close();
    }

    private void CancelClick(object sender, RoutedEventArgs e)
    {
        AuthorizationWindow aw = new AuthorizationWindow();
        aw.Show();
        this.Close();
    }
}
```

Изменим код в файле С# связанным с формой авторизации, а именно изменим код обработчика события клика по кнопке зарегистрироваться,

```
private void RegistrationClick(object sender, RoutedEventArgs e)
{
    RegistrationWindow rw = new RegistrationWindow();
    rw.Show();
    this.Close();
}
```

напишем там что бы он переходил на форму регистрации:

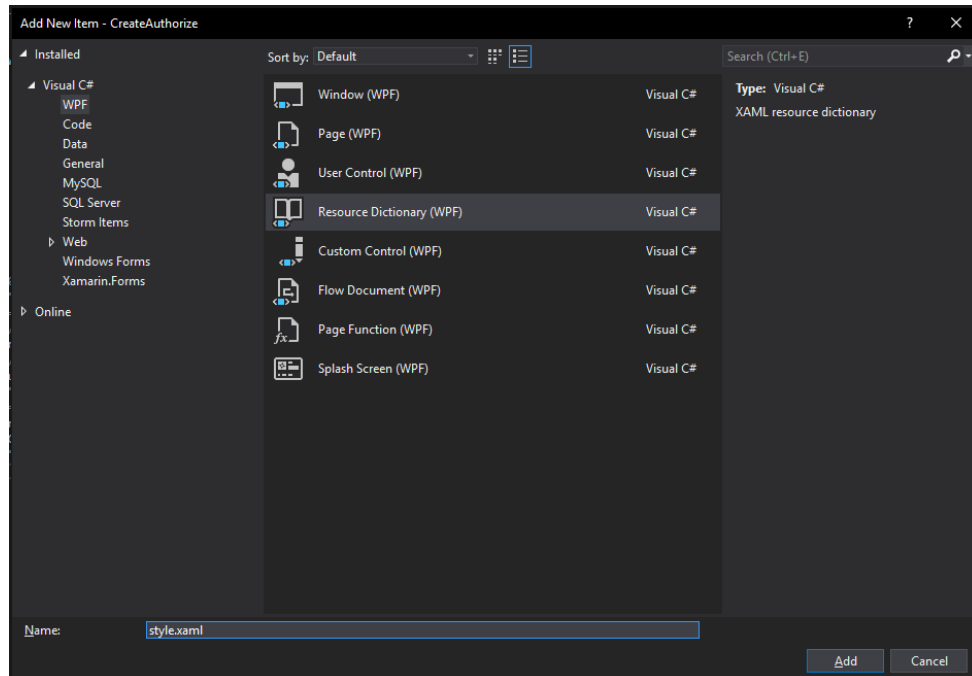
Теперь изменим файл App.xaml, чтобы запускалась форма авторизации.

2.3 Добавление стилей

```
<Application x:Class="CreateAuthorize.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:CreateAuthorize"
    StartupUri="AuthorizationWindow.xaml">
    <Application.Resources>
    </Application.Resources>
</Application>
```

Наше приложение работает, но выглядит не очень красиво, для того что бы сделать его красивее.

Для этого добавим такой элемент как словарь ресурсов, и дадим ему имя «Style.xaml»:



В открывшемся файле укажем нужные нам стили, они описаны ниже:

Тег Style говорит, что это стиль.

Тег Setter – обозначает свойство этого стиля.

Свойства:

TargetType – к кому мы применим стиль.

Property – наименование свойства которое хотим изменить.

Value – новое значение для выбранного свойства.


```

<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:CreateAuthorize">
    <Style TargetType="Button">
        <Setter Property="Margin" Value="5"/>
    </Style>
    <Style TargetType="TextBox">
        <Setter Property="HorizontalContentAlignment" Value="Center"/>
        <Setter Property="VerticalContentAlignment" Value="Center"/>
        <Setter Property="Margin" Value="5"/>
    </Style>
    <Style TargetType="PasswordBox">
        <Setter Property="HorizontalContentAlignment" Value="Center"/>
        <Setter Property="VerticalContentAlignment" Value="Center"/>
        <Setter Property="Margin" Value="5"/>
    </Style>
    <Style TargetType="TextBlock">
        <Setter Property="HorizontalAlignment" Value="Center"/>
        <Setter Property="VerticalAlignment" Value="Center"/>
        <Setter Property="Margin" Value="5"/>
    </Style>
</ResourceDictionary>

```

Теперь назначим файл «Style.xaml» основным стилем программы, для этого изменим файл App.xaml:

Здесь тег ResourceDictionary говорит, что мы хотим подключить словарь ресурсов, а свойство Source – обозначит источник, туда мы пишем имя файла, то есть пусть до него.

```

<Application x:Class="CreateAuthorize.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:CreateAuthorize"
    StartupUri="AuthorizationWindow.xaml">
    <Application.Resources>
        <ResourceDictionary Source="Style.xaml"/>
    </Application.Resources>
</Application>

```

Теперь наша программа работает.