



ChurchPoint

Church Management API and Church Management Website

Maintenance Manual

Product Owner

Pastor Pacifico Jun

Team Allegiant

Pix Mapanao | Stephen Gibson | Nathan Hidalgo | Jonathan Lopez

Emanuel Onofre | Keith Valdes | Antonio Castaneda

Version 1.0

May 14, 2022



ChurchPoint Software Solution |  ALLEGIANT.DEV

Table of Contents

1 Introduction	7
1 Comprehensive List of Technologies	8
1.1 Programming Languages	8
1.2 General Framework	8
1.3 Front-End Frameworks and Libraries	8
1.4 API Framework & Libraries	9
1.5 Back-End and Testing, Cloud Hosting Frameworks	9
1.6 Integrated Development Environment	9
1.7 Version Control System	9
2 Runbook	10
2.1 Server Goes Down	10
2.2 Database Becomes Corrupt	15
2.2.1 Using Azure when the Database Becomes Corrupt	15
2.2.2 Using the Development Environment to create a new database	15
2.3 Third Party Services Go Down	18
3 Configuring Development Environment	18
3 Configuring The Database	20
4 How to Make a Code Change	24
5 How to Examine Log Files and Errors	25
6 How to Deploy Releases	26
7 How to Run Automated Tests	28
8 Checks For Runtime Assurance	29
9 ChurchPoint Front-End Web Client	33
9.1 User Features	33
9.1.1 Registering a New Application User	33
9.1.2 Logging in with an Existing Application User	34



9.1.3 Logging out as an Application User	36
9.2 Basic Navigation	37
9.2.1 Accessing Home Page	37
9.2.2 Accessing Account Page	38
9.2.3 Accessing Churches Page	39
9.2.4 Accessing Attendees Page	41
9.2.5 Accessing Ministry Services Page	42
9.2.6 Accessing Donations Page	44
9.2.7 Accessing Events Page	46
9.2.8 Hiding/Showing the Navigation Bar	47
9.3 Church Features	49
9.3.1 Viewing Churches within the Database	49
9.3.2 Viewing Churches assigned to Application User	50
9.3.3 Viewing Church Detail Page	51
9.3.4 Paging: Setting Amount of Churches to View Within Church List Menu	53
9.3.5 Paging: Changing Between Pages Within the Church List Menu	54
9.3.6 Searching for Churches Using the Church List Menu	55
9.3.7 Creating New Churches	56
9.3.8 Temporarily Saving New Churches to be Added	57
9.3.9 Removing Churches From an Application User	59
9.3.10 Quick Editing Using the Church List Menu	60
9.3.11 Canceling Quick Editing Using the Church List Menu	61
9.3.12 Editing Using the Church Detail Page	62
9.4 Donations Features	63
9.4.1 Viewing Donations within the Database	63
9.4.2 Viewing Donations made by Application User	65
9.4.3 Paging: Setting amount of previous donations to view within Donations List Menu	66
9.4.4 Paging: Changing Between pages within the Donations List Menu	67
9.5 Attendees Features	68
9.5.1 Viewing Attendees within the Database	68



9.5.2 Viewing Attendees for Churches assigned to Application User	70
9.5.3 Paging: Setting amount of Attendees to view within the Attendees List Menu	71
9.5.4 Paging: Changing Between pages within the Attendees List Menu	72
9.6 Events Features	73
9.6.1 Viewing Events within the Database	73
9.6.2 Viewing Events for Churches assigned to Application User	75
9.6.3 Paging : Setting amount of Events to view within the Events List Menu	76
9.6.4 Paging : Changing Between pages within the Events List Menu	77
9.7 Ministry Service Features	78
9.7.1 Viewing Ministry Service within the Database	78
9.7.2 Viewing Ministry Service for Churches assigned to Application User	80
9.7.3 Paging: Setting amount of Ministry Service to view within the Ministry Service List Menu	81
9.7.4 Paging: Changing Between pages within the Ministry Service List Menu	82
10.0 ChurchPoint Web API	83
10.1 Open API Swagger Documentation Site General Features	83
10.1.1 Getting swagger.json document	83
10.1.2 Selecting API Definition	84
10.1.3 Looking at API Schemas	84
10.2 API Authorize Access w/ Bearer Token	85
10.2.1 Application User Authentication Register	85
10.2.2 Application User Authentication Authenticate	88
10.2.3 Authorize	89
10.3 Application User API Functions	92
10.3.1 Get Application User By Id	92
10.3.2 Update Application User	94
10.3.3 Delete Application User	96
10.3.4 Add Church To Application User	99
10.3.5 Remove Church From Application User	101
10.4 Attendees API Functions	103
10.4.1 Get Attendee By Id	103



10.4.2 Get Attendee By Church Id	105
10.4.3 Get Attendee By Event Id	108
10.4.4 Get Attendee By Ministry Service Id	110
10.4.5 Create Attendee User	112
10.4.6 Update Attendee User	115
10.4.7 Delete Attendee User	118
10.4.8 Remove Ministry Service From Attendee By Id	120
10.4.9 Remove Event From Attendee By Id	123
10.5 Churches API Functions	125
10.5.1 Get All Churches	125
10.5.2 Get Church By Id	127
10.5.3 Get All Churches By Application User Id	128
10.5.4 Get All Churches By Attendee User Id	130
10.5.5 Create Church User	132
10.5.6 Update Church User	134
10.5.7 Delete Church User	136
10.5.8 Add Attendee To Church By Id	139
10.5.9 Remove Attendee From Church By Id	141
10.6 Donation API Functions	143
10.6.1 Get All Donations	143
10.6.2 Get Donation By Id	145
10.6.3 Get All Donations By Attendee Id	148
10.6.4 Get All Donations By Church Id	150
10.6.5 Create Donation	152
10.6.6 Update Donation	155
10.6.7 Delete Donation	158
10.7 Event Features	160
10.7.1 Get All Events	160
10.7.2 Get Event By Id	162
10.7.3 Get All Events By Attendee Id	164



10.7.4 Get All Event By Ministry Service Id	166
10.7.5 Get All Events By Church Id	168
10.7.6 Create Event From Church	171
10.7.7 Create Event From Ministry Service	173
10.7.8 Update Event	176
10.7.9 Delete Event	179
10.7.10 Add Attendee To Event Id	181
10.7.11 Remove Attendee From Event Id	184
10.8 Ministry API Functions	187
10.8.1 Get All Ministry Services	187
10.8.2 Get Ministry By Id	189
10.8.3 Get Ministries By Church Id	191
10.8.4 Get Ministries By Attendee Id	193
10.8.5 Create Ministry	196
10.8.6 Update Ministry	197
10.8.7 Delete Ministry	200
10.8.8 Add Attendee to Ministry Service	202
10.8.9 Remove Attendee From Ministry Service	204
11 Database	207
Maintenance Manual Approval	208
Product Owner	208
The Maintenance Manual Document prepared by Allegiant.Dev	208

1 Introduction

The ChurchPoint WebAPI is a fully functional RESTful application programming interface that is made for servicing multiple clients ranging from websites, phone apps, desktop apps, etc. Functionally works as a gateway for services that are primarily dealing with the persistence of Church managerial data such as Attendees, Ministry Services, Donations, and Events.

This document contains the necessary steps to successfully maintaining the ChurchPoint Software Solution. Below are the main sections the document contains.

This Maintenance Document includes the following

1. Comprehensive List of Technologies Used
2. Runbook
 - a. Server Goes Down
 - i. Restart / Recovery
 - b. Database becomes corrupt
 - c. Third Party Services Go Down
 - i. Backup Strategy
 - ii. Errors Seen
3. Configuring Development Environment
4. How to make a code change
5. How to examine log files for errors
6. How to deploy releases
7. How to run automated tests
8. Checks for Runtime Assurance
9. User Interaction Front-End Webpage
 - a. Screenshots
 - b. Description
 - c. List of code files related to Interaction
10. User Interaction Back-End (API, Application Core, Infrastructure)
 - a. Screenshots
 - b. Description
 - c. List of code files related to interaction
11. Database
12. Appendices



1 Comprehensive List of Technologies

The Technology stack used to develop this project is primarily the **Microsoft Technical Stack**.

For further information on this visit:



<https://docs.microsoft.com/en-us/documentation/>

1.1 Programming Languages

Database Language : T-SQL

[https://docs.microsoft.com/en-us/sql/t-sql/language-reference?
view=sql-server-ver15](https://docs.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver15)



Back-End Language: C#

<https://docs.microsoft.com/en-us/dotnet/csharp/>

Front-End Languages: Razor, HTML5, CSS3, JavaScript

<https://docs.microsoft.com/en-us/aspnet/core/mvc/views/razor?view=aspnetcore-6.0>

1.2 General Framework



General Framework : .NET Core 6

<https://docs.microsoft.com/en-us/dotnet/core/whats-new/dotnet-6>

1.3 Front-End Frameworks and Libraries

Front-End Framework : Blazor

Front-End CSS Libraries: Bootstrap , Material Design, MudBlazor



<https://docs.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-6.0>

<https://getbootstrap.com/docs/5.1/getting-started/introduction/>

<https://mdbbootstrap.com/docs/standard/>

<https://www.mudblazor.com/docs/overview>



1.4 API Framework & Libraries

API Framework : OpenAPI



API Library : Swagger

<https://oai.github.io/Documentation/>

<https://swagger.io/>

1.5 Back-End and Testing, Cloud Hosting Frameworks

Database Technologies : Microsoft SQL Server



Hosting Framework : Microsoft Azure



Testing Frameworks : MSTest , Selenium



<https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15>

<https://docs.microsoft.com/en-us/azure/?product=popular>

<https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-mstest>

<https://docs.microsoft.com/en-us/azure/devops/pipelines/test/continuous-test-selenium?view=azure-d-evops>

1.6 Integrated Development Environment

Integrated Development Environment : Visual Studio 2022



<https://docs.microsoft.com/en-us/visualstudio/windows/?view=vs-2022>

1.7 Version Control System

Version Control System : GitHub



<https://docs.github.com/en>

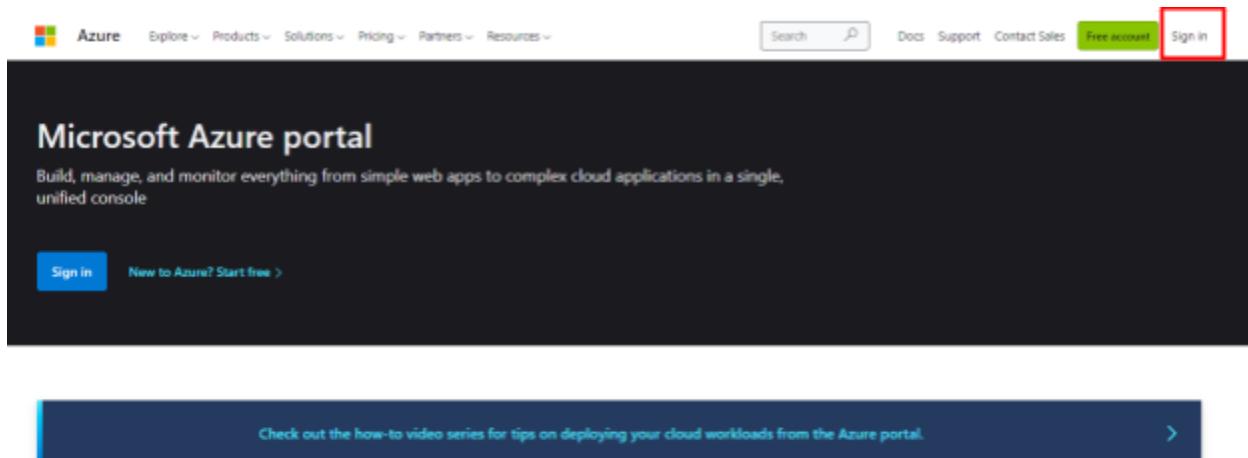
2 Runbook

2.1 Server Goes Down

All server hosting for ChurchPoint Software Solution is done on Microsoft's Azure. In the event of server failure, you will need to login to the administrator azure account which is disclosed to the Product Owner.

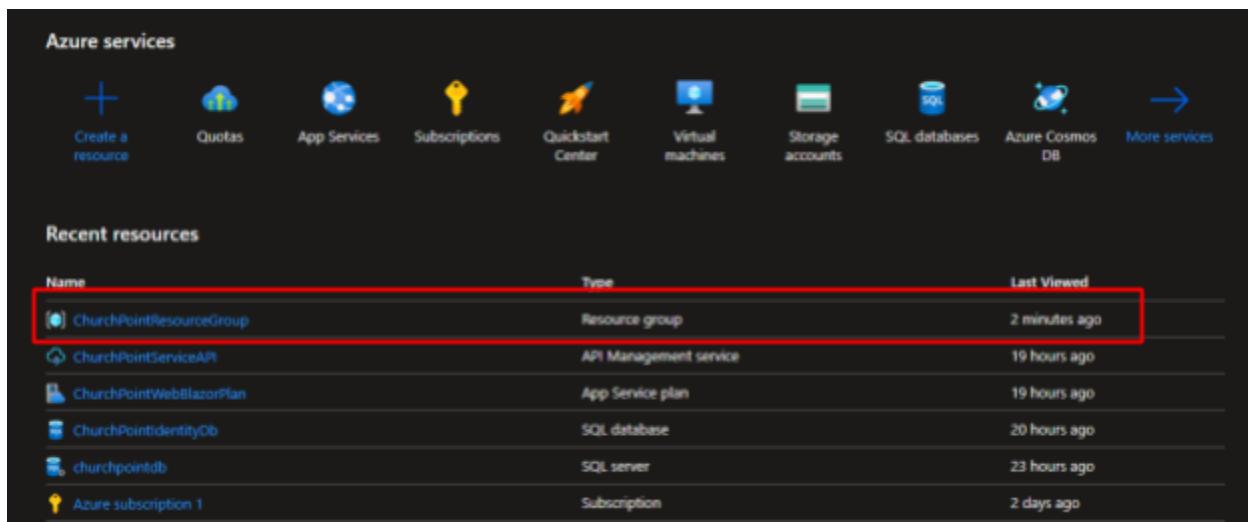
The steps are as follows:

1. Navigate to <https://azure.microsoft.com/en-us/features/azure-portal/> and click 'Sign in'



The screenshot shows the Microsoft Azure portal homepage. At the top, there is a navigation bar with links for Azure, Explore, Products, Solutions, Pricing, Partners, Resources, a search bar, and buttons for Docs, Support, Contact Sales, Free account, and Sign in. The 'Sign in' button is highlighted with a red box. Below the navigation bar, the title 'Microsoft Azure portal' is displayed, followed by a subtitle: 'Build, manage, and monitor everything from simple web apps to complex cloud applications in a single, unified console.' There are two buttons at the bottom left: 'Sign in' and 'New to Azure? Start free >'. A banner at the bottom of the page says 'Check out the how-to video series for tips on deploying your cloud workloads from the Azure portal.' with a right-pointing arrow.

2. Then from the Azure Home Screen, navigate to ChurchPointResourceGroup



The screenshot shows the Azure Home screen. At the top, there is a section titled 'Azure services' with icons for Create a resource, Quotas, App Services, Subscriptions, Quickstart Center, Virtual machines, Storage accounts, SQL databases, Azure Cosmos DB, and More services. Below this, there is a section titled 'Recent resources' with a table. The table has columns for Name, Type, and Last Viewed. The first row, 'ChurchPointResourceGroup', is highlighted with a red box. The table data is as follows:

Name	Type	Last Viewed
ChurchPointResourceGroup	Resource group	2 minutes ago
ChurchPointServiceAPI	API Management service	19 hours ago
ChurchPointWebBlazorPlan	App Service plan	19 hours ago
ChurchPointIdentityDb	SQL database	20 hours ago
churchpointdb	SQL server	23 hours ago
Azure subscription 1	Subscription	2 days ago

3. From the Resource Group, cycle through each element of the group's services and check the Status of all resources. Making sure it says 'Ready' or 'Running', below is the example of the API



Home > ChurchPointResourceGroup >

ChurchPointAPI

App Service plan

Search (Ctrl+ /) | Delete | Send us your feedback

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Events (preview)

Status : Ready

Location : West US

Subscription (move) : Azure subscription 1

Subscription ID : 6f677cae-0b49-4b26-83fb-cbf6b9200a27

Tags (edit) : Click here to add tags

ChurchPointAPI

App Service

Search (Ctrl+ /) | Browse | Stop | Swap | Restart | Delete |

Click here to access Application Insights for monitoring and profiling fo

Overview

- Activity log
- Access control (IAM)
- Tags
- Diagnose and solve problems
- Security
- Events (preview)

Status : Running

Location : West US

Subscription (move) : Azure subscription 1

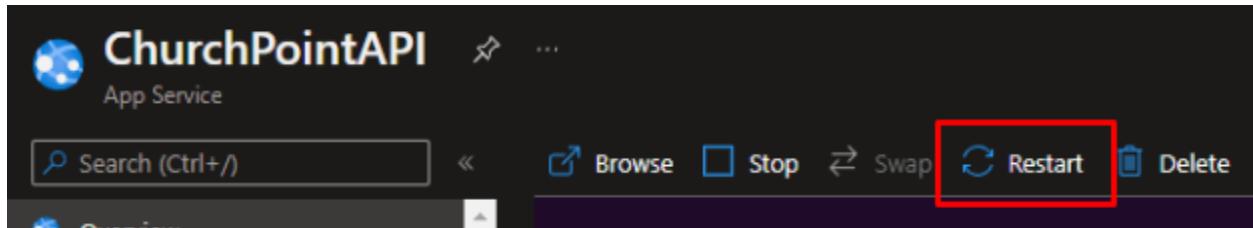
Subscription ID : 6f677cae-0b49-4b26-83fb-cbf6b9200a27

Tags (edit) : Click here to add tags

4. Repeat Step 3 for all resource items especially the Blazor Web Page hosting resource item and check for a Status ‘Running’ or ‘Ready’



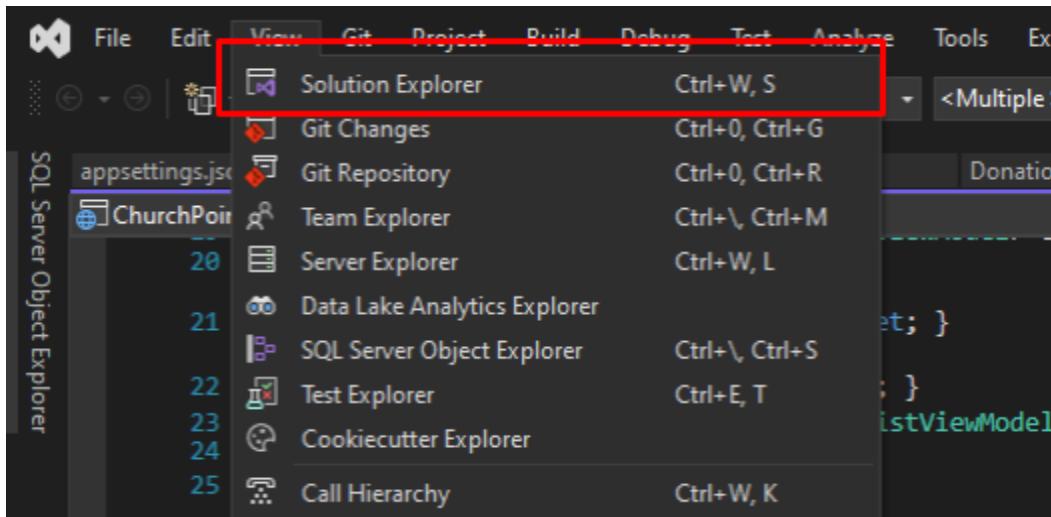
- Once checks are made, you can click on the 'Restart' service for any non-running or non-ready instances.



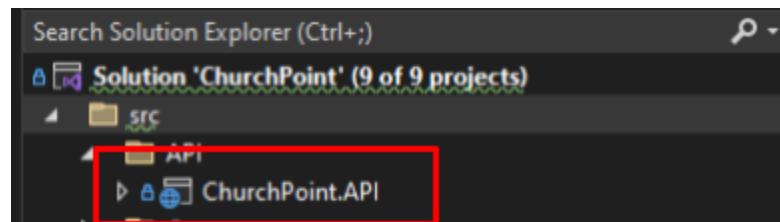
- If it is unable to switch back to a running or ready status **contact Azure support** found on the bottom of your sidebar.

A screenshot of the Microsoft Azure portal showing the 'ChurchPointAPI' App Service overview. The left sidebar shows navigation links like Home, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Security, Events (preview), Deployment, Quickstart, Deployment slots, Deployment Center, Settings, Configuration, Authentication, Application Insights, Identity, and Backups. The 'Identity' link is highlighted with a red box. The main content area displays the app's status as 'Running', location as 'West US', and subscription information. It also includes sections for 'Diagnose and solve problems' and 'Application Insights'.

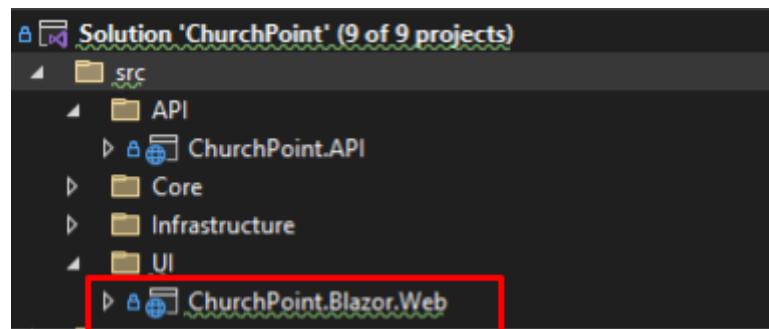
- Additionally, if you'd like to continue diagnosing the problem yourself you can go to **Section 3 of this document to configure the Development Environment** and Re-deploy the application. In order to do **follow all necessary steps in Section 3 for development environment configuration**, then navigate to the Solution Explorer.



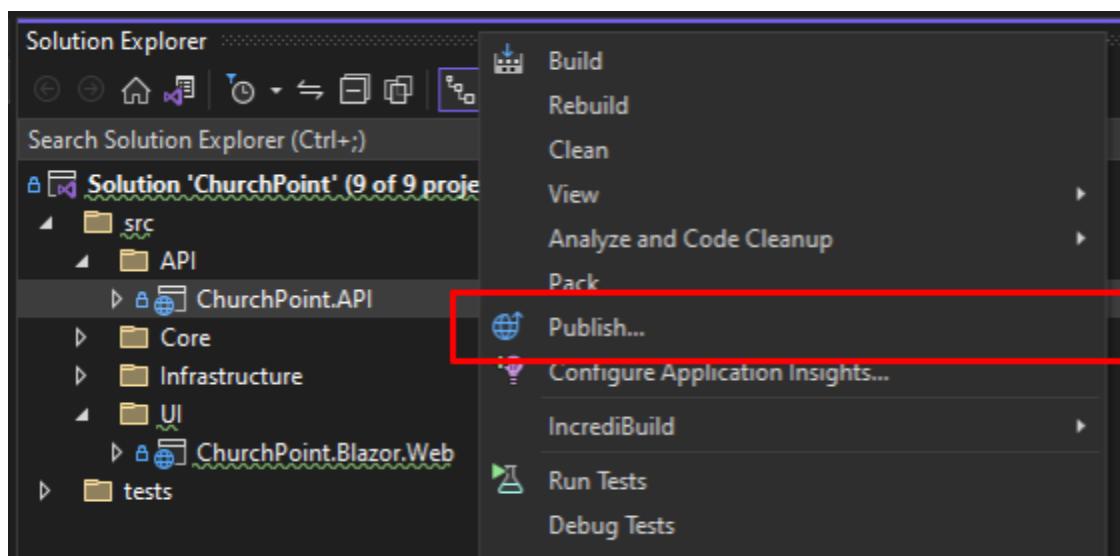
8. After navigating to the Solution Explorer, open: src → API and you will see the 'ChurchPoint.API' project.



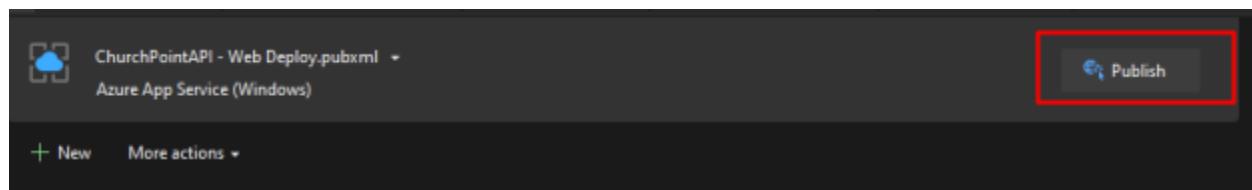
9. Also, you will need to open the UI folder and you will see the 'ChurchPoint.Blazor.Web' project.



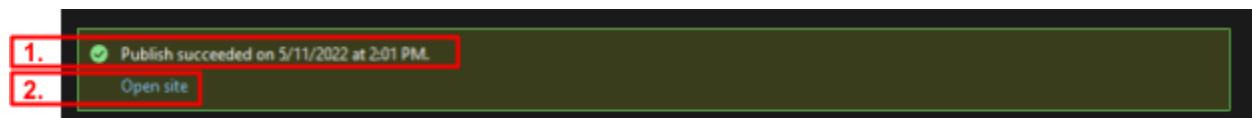
10. Then for each project, right click it and hit 'publish'. Below is an example on the API project.



11. Then on the publish window hit 'publish'. This should rebuild the project and redeploy it onto Azure.



12. Make sure that you get a publish success prompt after publishing and open the website to verify.



13. You should now see the published website. If you are still unable to get a success prompt or other problems persist. Go to Step 6 to contact Azure Support, or contact the Project Manager at the email : pmapanao@csus.edu for further assistance.



2.2 Database Becomes Corrupt

In the event that the database becomes corrupt you will need to follow the set-up in **Section 3 Configuring Development Environment** for the development solution for this fix. There will be two main solutions:

1. Use Azure to revert back to an earlier state of the Database
2. Use the Development Environment to create a new database

2.2.1 Using Azure when the Database Becomes Corrupt

Follow the steps 1 - 2 in **Section 2.1 Server goes down** before proceeding

1. After following Steps 1-2 in Section 2.1, click on the ChurchPointIdentityDb stored on the cloud

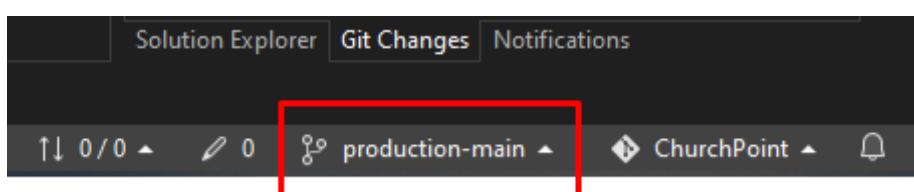
Name	Type	Location
ChurchPointAPI	App Service plan	West US
ChurchPointAPI	App Service	West US
ChurchPointIdentity	App Service	West US
ChurchPointIdentityWeb	SignRift	West US
ChurchPointIdentityWebCI	SQL server	West US
churchpointdb	SQL database	West US
ChurchPointIdentityDb (churchpointdb.ChurchPointIdentityDb)	SQL database	West US
ChurchPointIdentity	Job Management service	West US
ChurchPointIdentityWeb	App Service	West US
ChurchPointIdentityWebCI	App Service plan	West US

2. Once in the status menu click Restore, then follow this link for more detailed instructions on restoring the Database through Azure Hosting Services:
<https://docs.microsoft.com/en-us/azure/azure-sql/database/recovery-using-backups?view=azuresql>

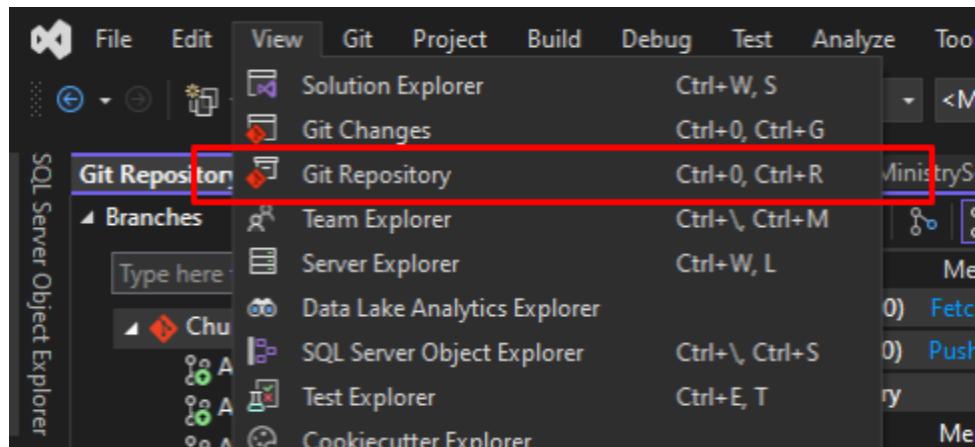
2.2.2 Using the Development Environment to create a new database

Follow the steps in **Section 3 Configuring Development Environment** before proceeding with this fix

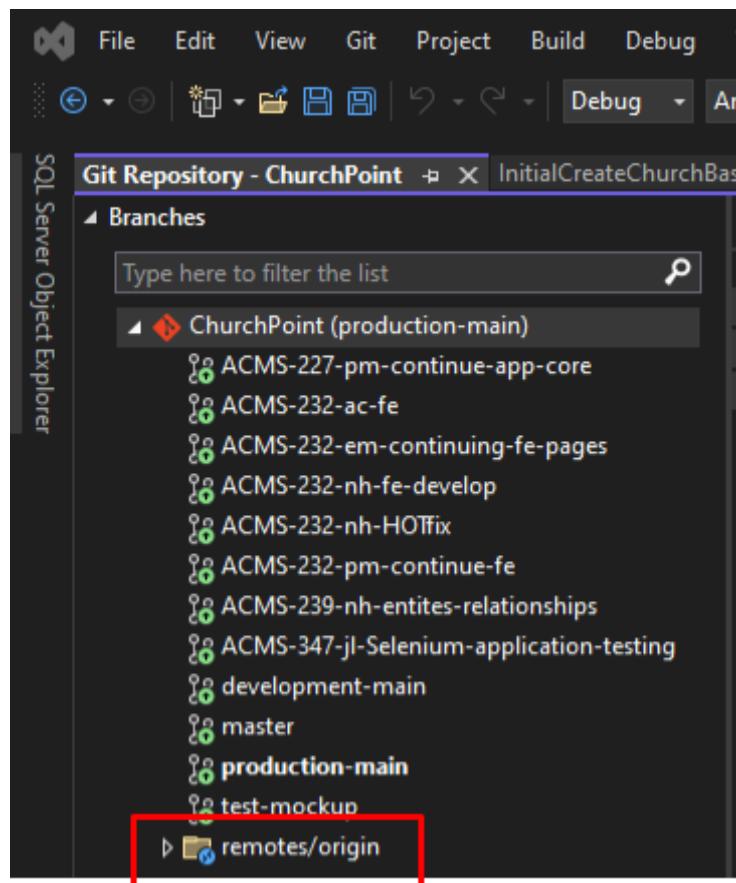
1. In the case of catastrophic failure, the development environment can be used to recreate the database from scratch. In order to get previous data that was stored on the database, scheduled backups need to be stored locally on the server machine. You can check this through Azure. Once retrieving the .bak file of the Db, it is possible to do a system restore with that seed.
2. After configuring the development environment, you must first be on the production branch labeled 'production-main'.



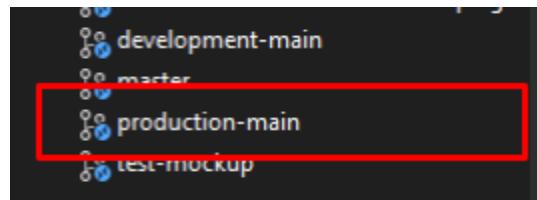
3. In order to get to this branch, you will need to go to View → Git Repository on Visual Studio 2022



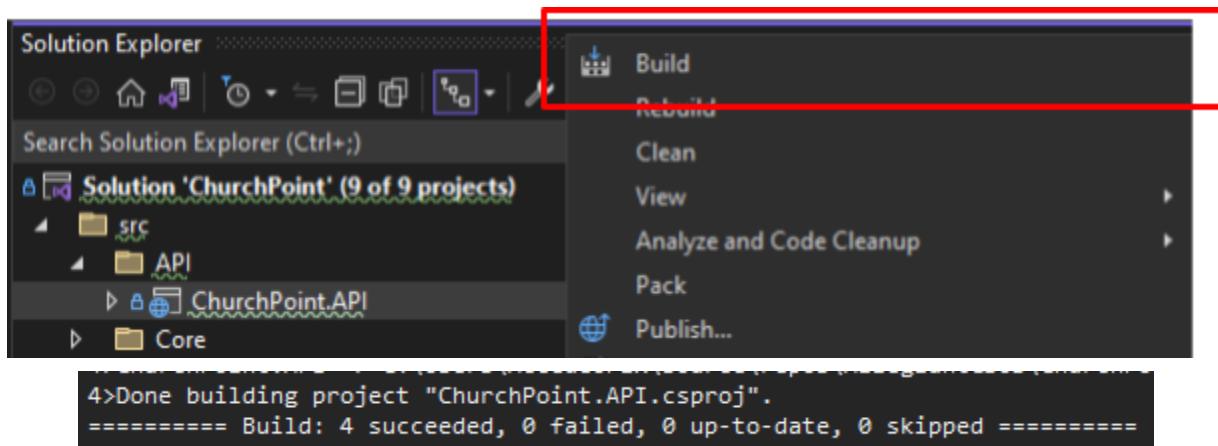
4. Once in the Git Repository Menu, expand the remotes/origin folder found on the left hand side.



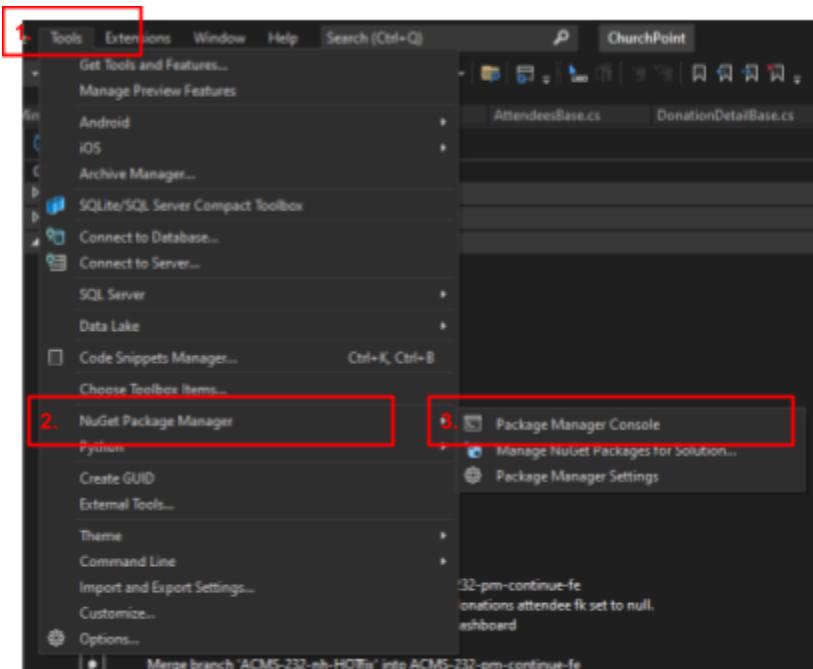
5. After expanding it, look for the ‘production-main’ branch and double click it to pull it locally



6. After pulling it locally, go to the Solution explorer and target the ChurchPoint.API project then right-click and build the project



7. After the successful build of the project, open up the package manager. You can do this by going to Tools → NuGet Package Manager → Package Manager Console



- Finally, Refer to the set-up **Database Creation part of Section 3** for to re-create the database through the NuGet Package Manager Console

2.3 Third Party Services Go Down

In the event that the third party hosting website Azure goes down you will need to be aware of the following information:

- It is **highly important** to always maintain backups and pay attention to Emails regarding services going down.
- In the event Azure goes down, though unlikely, you will need to download your database data in order to restore the existing database data on production
- Make sure you follow the steps for **Section 3 Configuring Development Environment** so you can publish to Publish the ChurchPoint Software Solution to the IIS (Internet Information Services) in order to deploy as a Web Package
- More information on the starting steps to deploy into the IIS will be found on **Section 6 How to Deploy Releases**

The course of action when Third Party Services go down is to:

- Gather all the backups from Azure before the Third Party Service goes down by paying attention to any information regarding the Azure deployment found on the admin account with credentials given to the Product Owner.
- Next, is to do the steps in **Section 3 Configuring Development Environment** to set-up a Development environment for redeployment.
- Finally, follow the correct database restoration procedures found in **Section 3 Configuring the Development Environment**

3 Configuring Development Environment

In order to configure the Development Environment please follow these steps:

- Make sure to have the **.NET 6 Software Development Kit** downloaded and installed on your computer. The computer needs to be running on the **Windows OS**. Other operating systems may have compatibility, but to ensure best results a Windows machine is needed. More detailed instructions on this process is well documented on Microsoft and can be found in the following link:

<https://docs.microsoft.com/en-us/dotnet/fundamentals/>

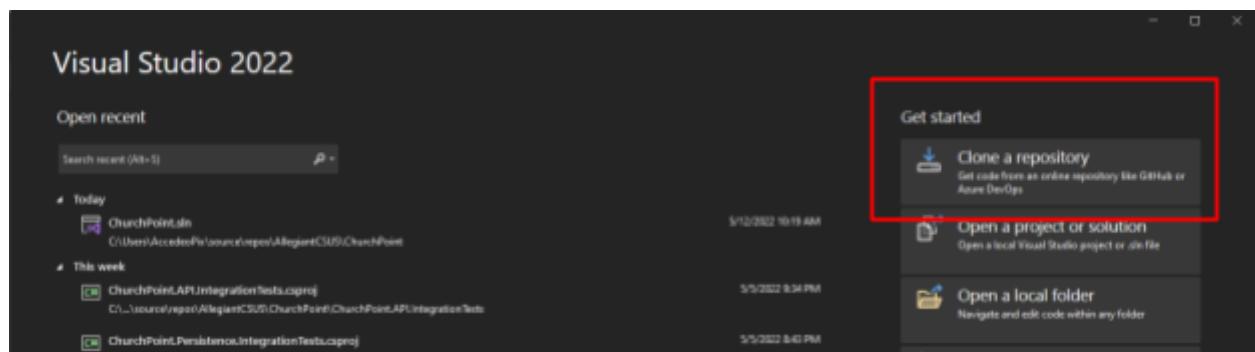


2. After getting the .NET 6 downloaded and installed, make sure to download **Visual Studio 2022** and configure your development environment through their straightforward installer.

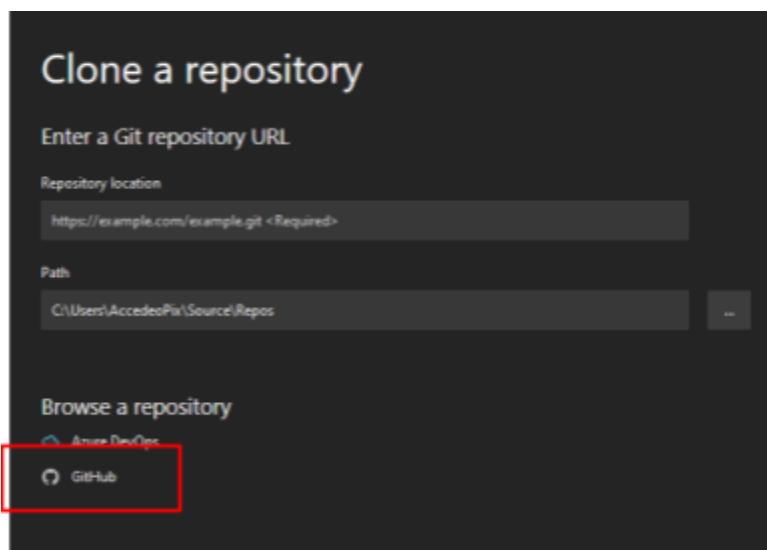
Documentation on this process is well documented on Microsoft and can be found with the following link.

<https://visualstudio.microsoft.com/vs/getting-started/>

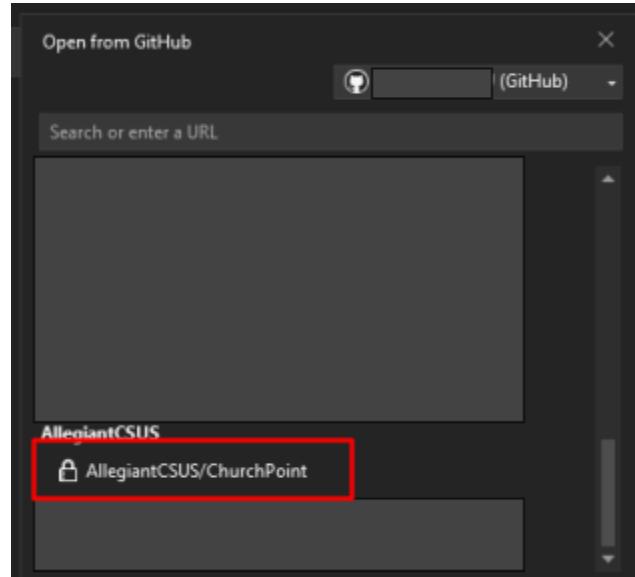
3. After getting Visual Studio 2022 installed, we can now set up our access to **GitHub**
4. Make sure to set-up an account with **GitHub** and the process is well documented with the link shown below.
<https://docs.github.com/en/get-started>
5. Once a User Account is set up with GitHub, now we can open **Visual Studio 2022**. Then hit the 'Clone a Repository' button.



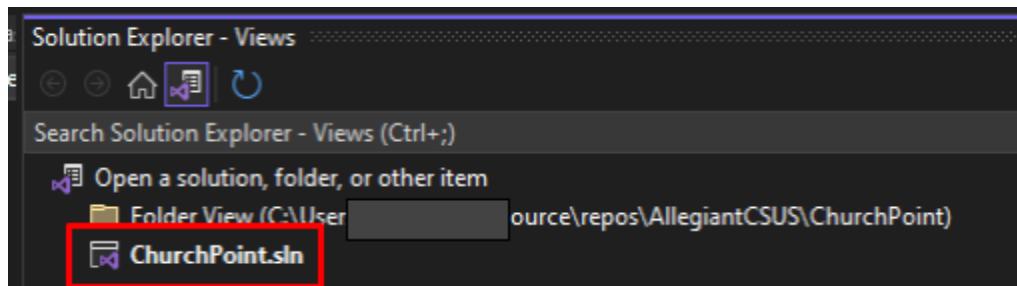
6. Then you will hit the GitHub button found on the bottom of the next menu



- After hitting the GitHub button, you will see a list of Repositories make sure to click on the **AllegiantCSUS/ChurchPoint** option



- After that, Visual Studio will download the Repository and Create your Development Environment. If you are seeing a folder view on your Solution Explorer, make sure to search for the solution file and double-click it.



- The Development Environment is now nearly set up, next we will talk about Configuring the Database.

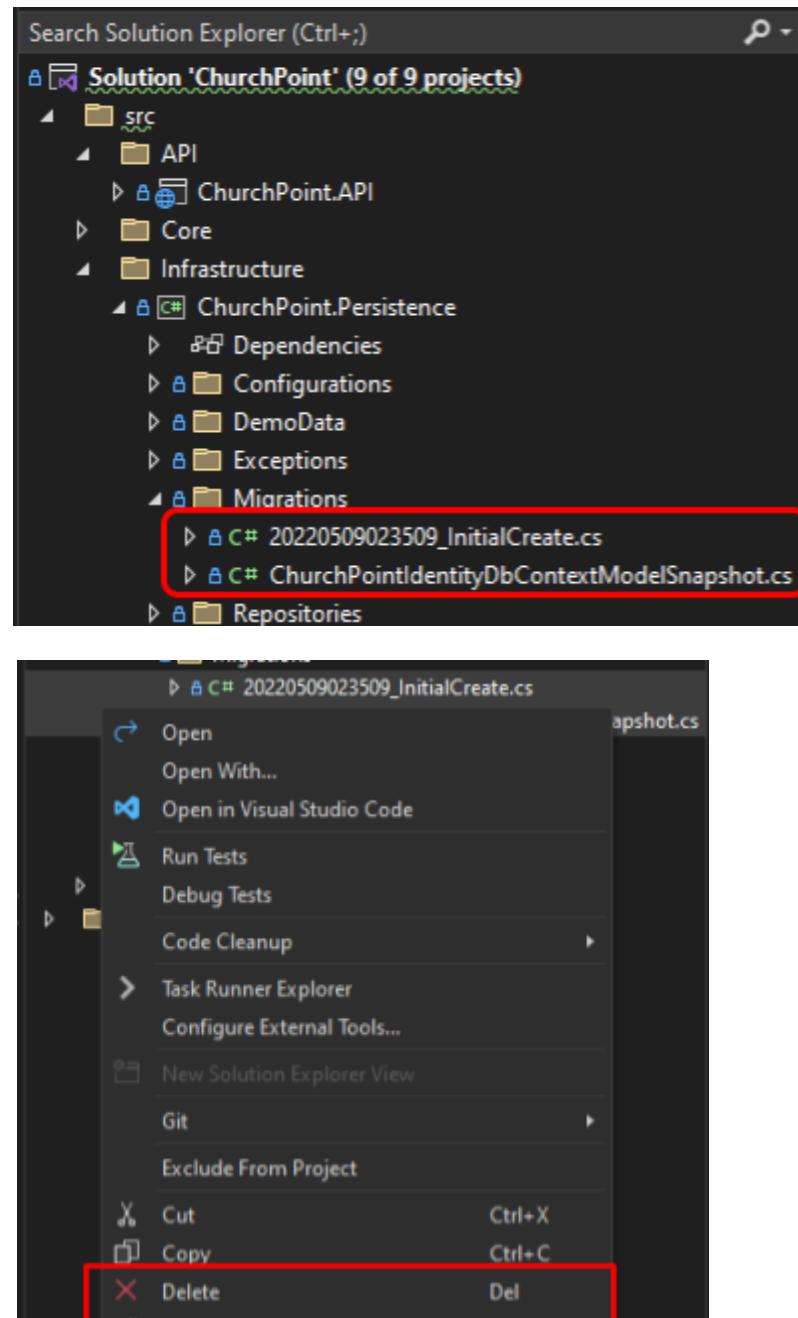
3 Configuring The Database

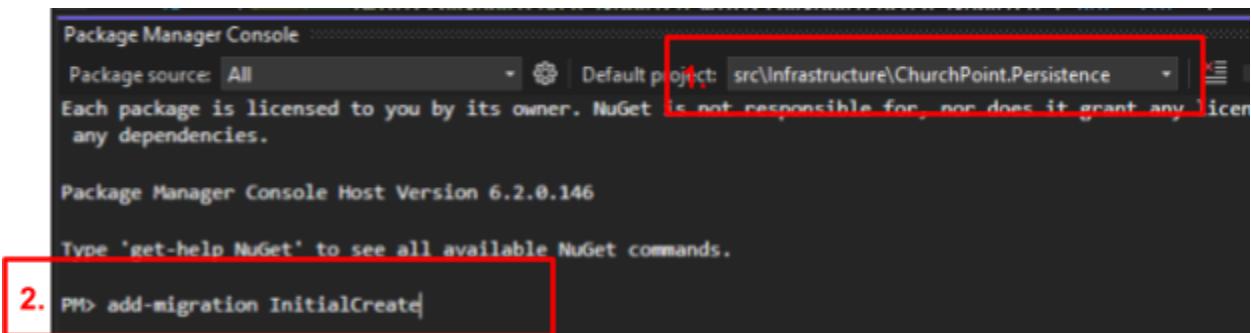
In order to configure the Database for your Development Environment you need to do all the preliminary steps found in **Section 3 Configuring Development Environment**

- After setting up, follow the instructions on **Section 2.2.2 Using Development Environment to Create a New Database EXCEPT USE THE 'development-main' BRANCH**, then refer back to here after the Package Manager step.

2. Within the Package Manager Console type in ‘**add-migration InitialCreate**’

Note: If there is anything in the Migrations Folder under Src → Infrastructure → Migrations, delete them. To do that Click shift the two files and then right-click and press delete.





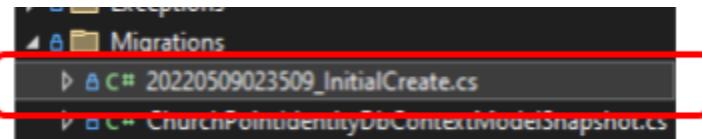
```
Package Manager Console
Package source: All | Default project: src\Infrastructure\ChurchPoint.Persistence | X
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any license to, any dependencies.

Package Manager Console Host Version 6.2.0.146
Type 'get-help NuGet' to see all available NuGet commands.

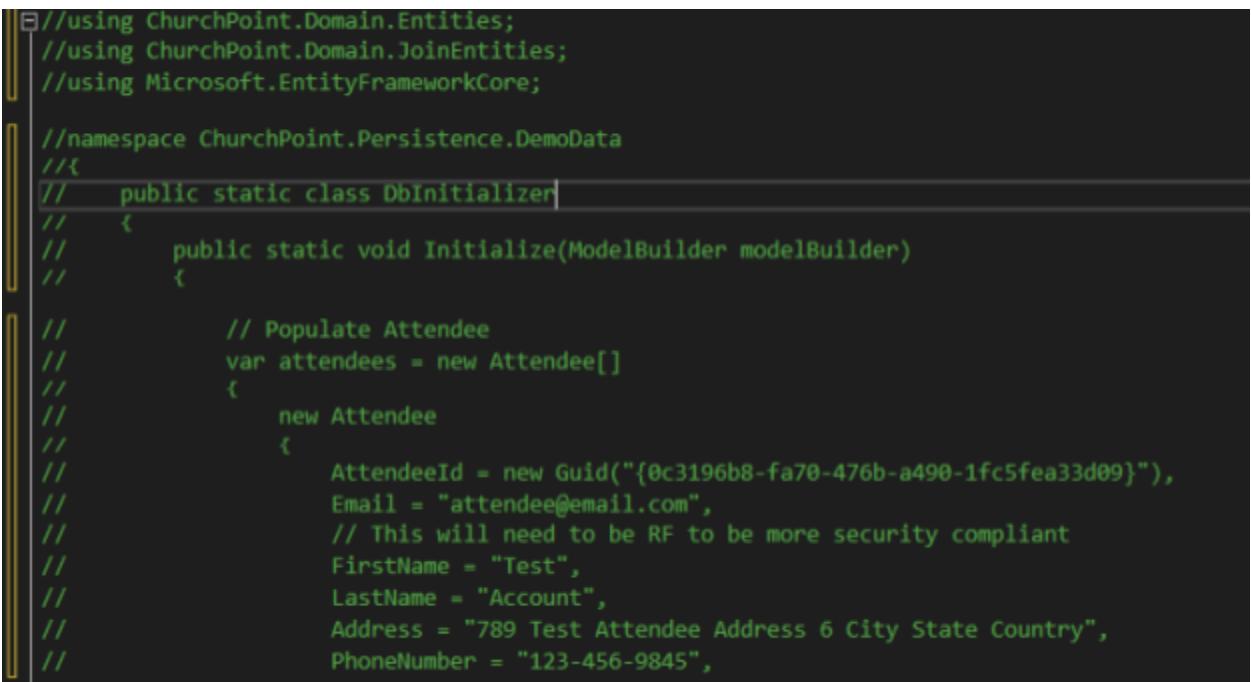
2. PM> add-migration InitialCreate
```

3. After letting the add-migration script run, make sure that there is a Migration found on the Migrations folder.

Note: If you don't want the seed data to populate make sure to comment out all code in the DbInitializer.cs file found in the DemoData folder.



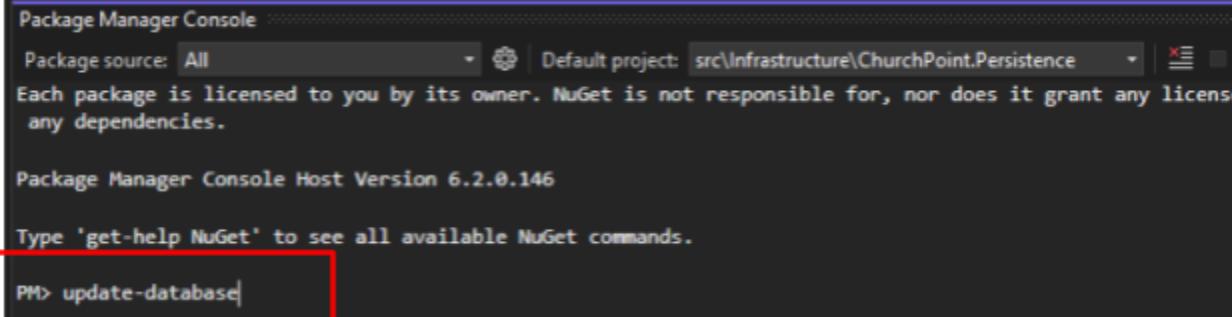
To comment out the code, simply press Ctrl+A, K + C



```
//using ChurchPoint.Domain.Entities;
//using ChurchPoint.Domain.JoinEntities;
//using Microsoft.EntityFrameworkCore;

//namespace ChurchPoint.Persistence.DemoData
//{
//    public static class DbInitializer
//    {
//        public static void Initialize(ModelBuilder modelBuilder)
//        {
//
//            // Populate Attendee
//            var attendees = new Attendee[]
//            {
//                new Attendee
//                {
//                    AttendeeId = new Guid("{0c3196b8-fa70-476b-a490-1fc5fea33d09}"),
//                    Email = "attendee@email.com",
//                    // This will need to be RF to be more security compliant
//                    FirstName = "Test",
//                    LastName = "Account",
//                    Address = "789 Test Attendee Address 6 City State Country",
//                    PhoneNumber = "123-456-9845",
//                    ZipCode = "12345"
//                }
//            };
//        }
//    }
//}
```

4. Next, go to the Package Manager Console again and type in '**update-database**' to create the database locally



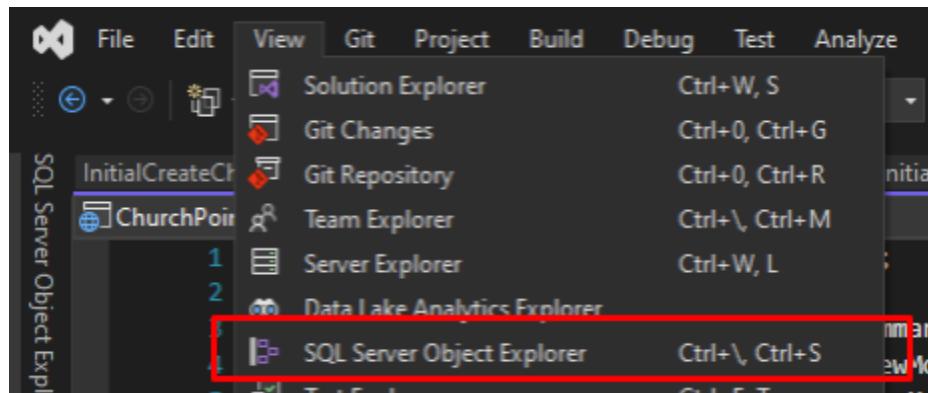
```
Package Manager Console
Package source: All | Default project: src\Infrastructure\ChurchPoint.Persistence | X
Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any license
any dependencies.

Package Manager Console Host Version 6.2.0.146

Type 'get-help NuGet' to see all available NuGet commands.

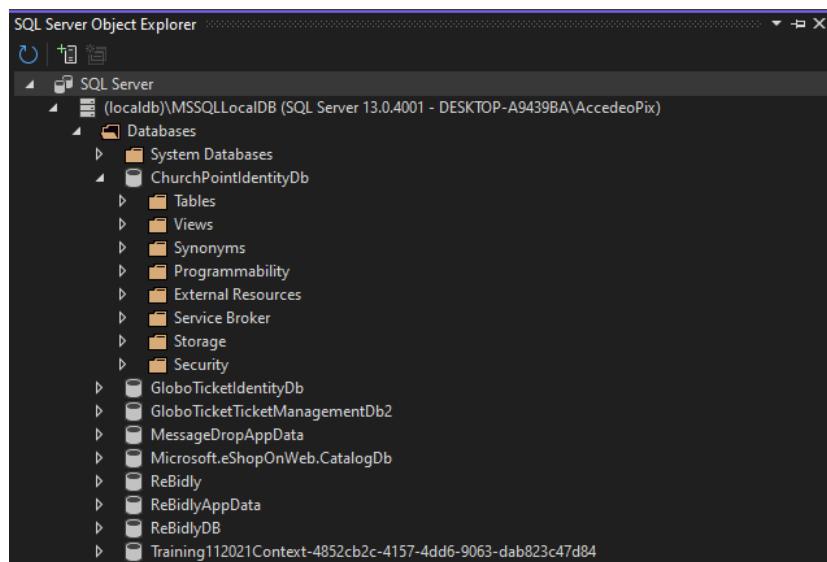
PM> update-database
```

5. After the Database Script runs, open up the SQL Object Explorer

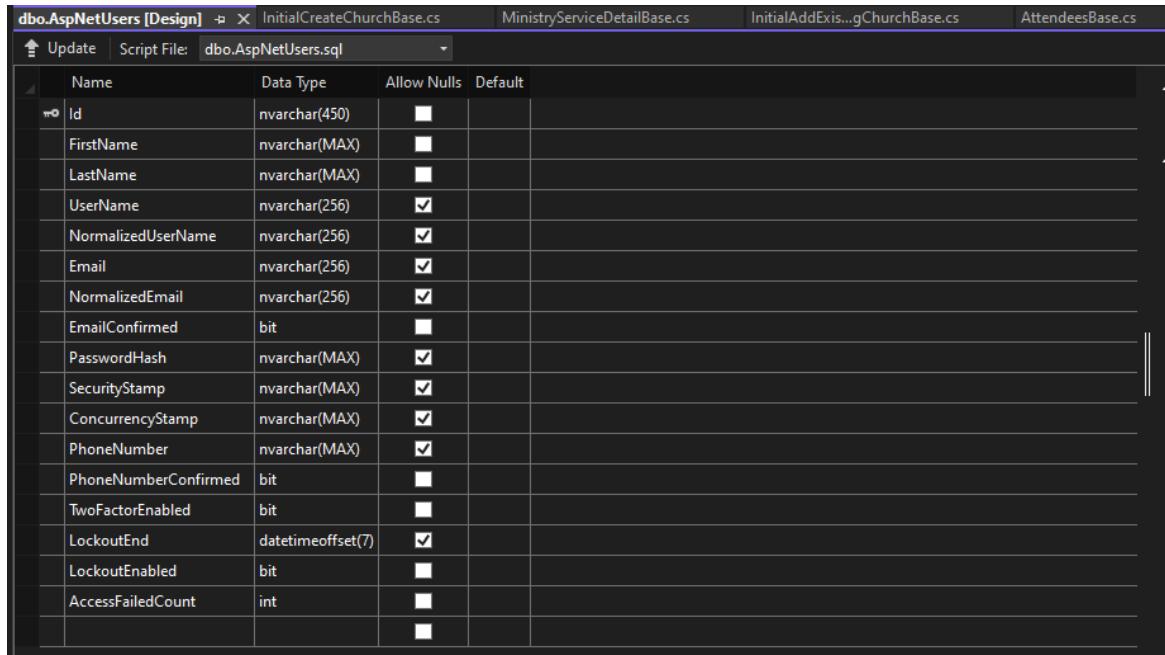


6. Then on the (localdb)\MSSQL Instance expand until you see the ChurchPointIdentityDb confirming the database is created locally.

NOTE: If you are doing this on the Azure database make sure you are on the '**production-main**' branch to update the Azure Database or if you're using another hosting site you should be able to view the connection by adding the server on the top of the SQL Object Explorer



7. Finally to confirm you can expand the tables section and click any of the tables to see their schema.



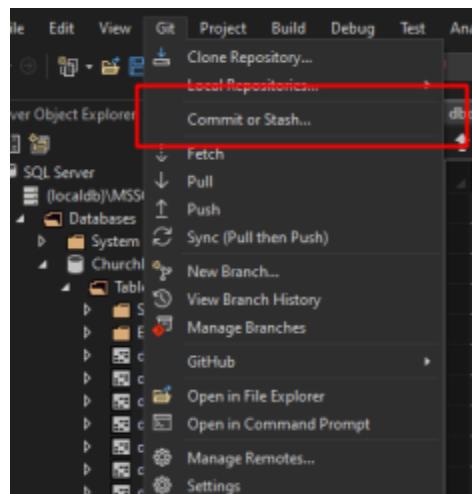
The screenshot shows the SQL Server Object Explorer with the 'Tables' node expanded. A red box highlights the 'dbo.AspNetUsers' table. The table has 15 columns: Id, FirstName, LastName, UserName, NormalizedUserName, Email, NormalizedEmail, EmailConfirmed, PasswordHash, SecurityStamp, ConcurrencyStamp, PhoneNumber, PhoneNumberConfirmed, TwoFactorEnabled, LockoutEnd, LockoutEnabled, and AccessFailedCount. The 'Allow Nulls' column contains several checked boxes, indicating that null values are allowed for those specific columns.

Name	Data Type	Allow Nulls	Default
Id	nvarchar(450)	<input type="checkbox"/>	
FirstName	nvarchar(MAX)	<input type="checkbox"/>	
LastName	nvarchar(MAX)	<input type="checkbox"/>	
UserName	nvarchar(256)	<input checked="" type="checkbox"/>	
NormalizedUserName	nvarchar(256)	<input checked="" type="checkbox"/>	
Email	nvarchar(256)	<input checked="" type="checkbox"/>	
NormalizedEmail	nvarchar(256)	<input checked="" type="checkbox"/>	
EmailConfirmed	bit	<input type="checkbox"/>	
PasswordHash	nvarchar(MAX)	<input checked="" type="checkbox"/>	
SecurityStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>	
ConcurrencyStamp	nvarchar(MAX)	<input checked="" type="checkbox"/>	
PhoneNumber	nvarchar(MAX)	<input checked="" type="checkbox"/>	
PhoneNumberConfirmed	bit	<input type="checkbox"/>	
TwoFactorEnabled	bit	<input type="checkbox"/>	
LockoutEnd	datetimeoffset(7)	<input checked="" type="checkbox"/>	
LockoutEnabled	bit	<input type="checkbox"/>	
AccessFailedCount	int	<input type="checkbox"/>	
		<input type="checkbox"/>	

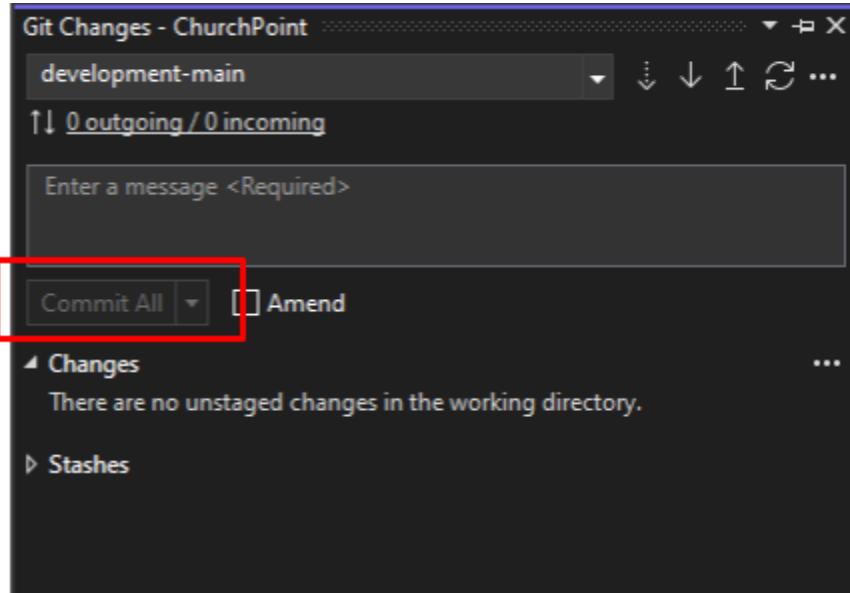
4 How to Make a Code Change

In order to make a code change make sure to follow the **Section 3 Configuring the Development Environment** and **Section 3 Configuring the Database**, then do the following steps:

1. Open up Visual Studio 2022
2. Then, click on any file you want to edit
3. Write your edits on the file and save
4. Then if you want to persist the changes go to Git → Commit or Stash



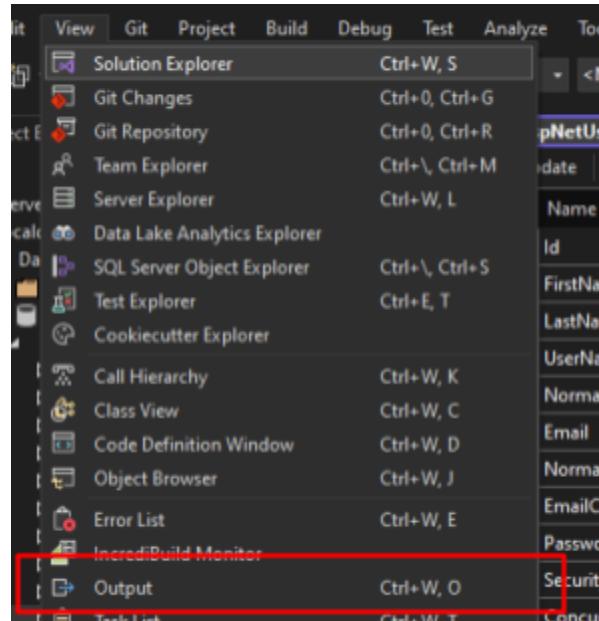
- Then when on the Git Changes Menu, review your change thoroughly per file them press commit all.

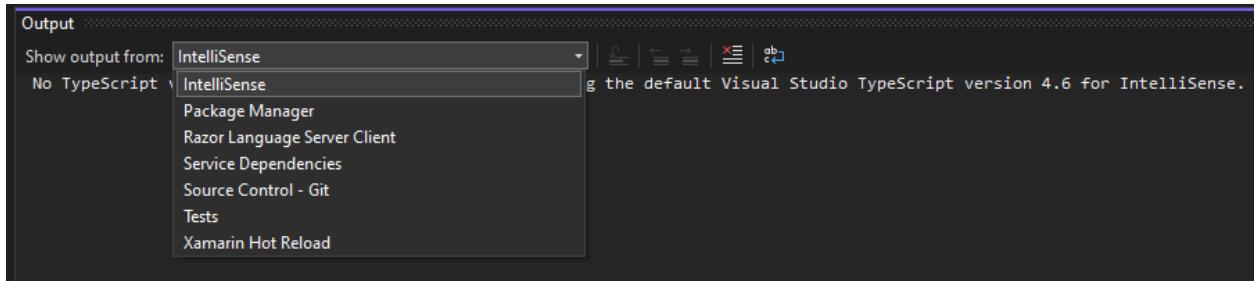


- Then press the Up Button to push it all up to the GitHub Repository.

5 How to Examine Log Files and Errors

In order to examine the Log Files and Errors you can check anything on the build using the Output menu on Visual Studio for any immediate things being logged to Console out. In order to do this you can go to View → Output, in here all the immediate actions will be logged, seen below:





Additional logs and resources can be found through **Visual Studio 2022** as well. In order to access these resources go through the Microsoft Documentation following the link below:

<https://docs.microsoft.com/en-us/visualstudio/ide/how-to-view-save-and-configure-build-log-files?view=vs-2022>

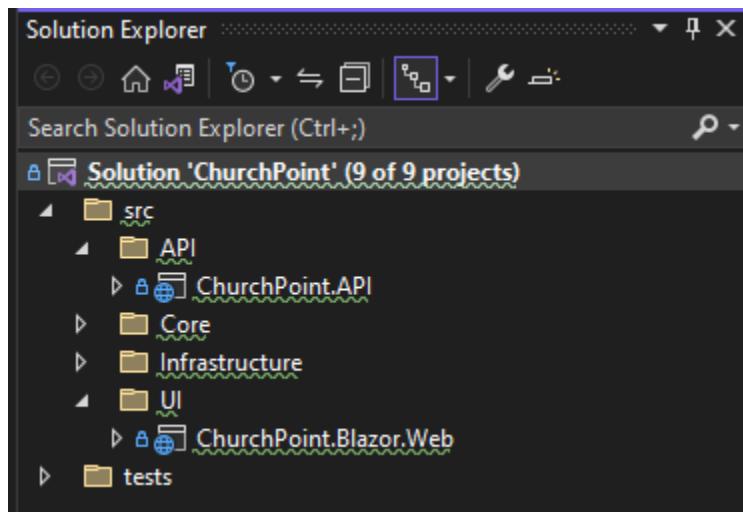
6 How to Deploy Releases

In order to deploy releases please follow these steps:

To deploy Releases on Azure follow **Section 2.1 When Server Goes down from Step 7 and onwards** to deploy Releases to Azure and **Section 3 Configuring the Development Environment and Database**

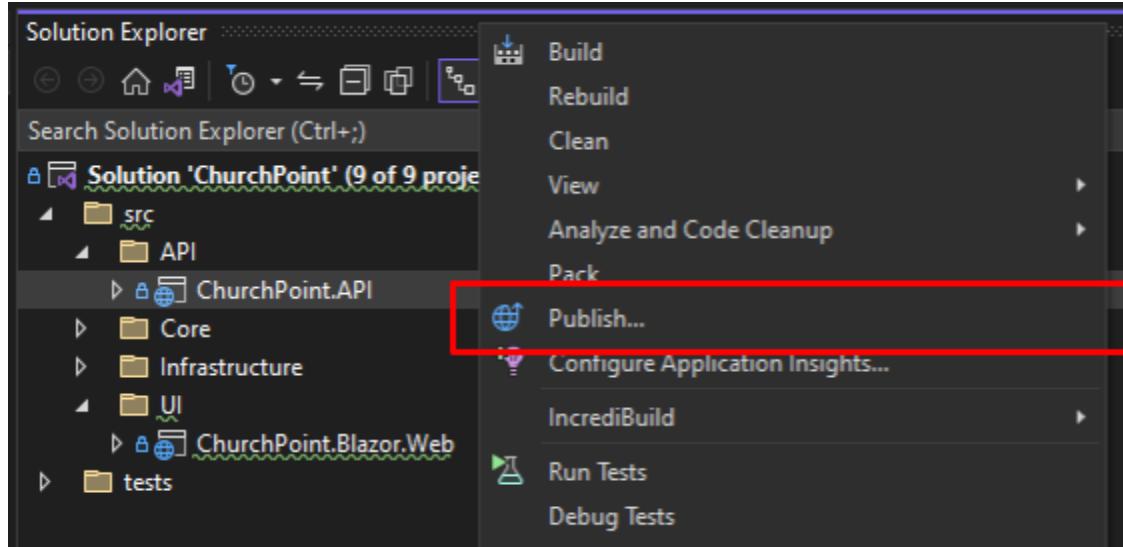
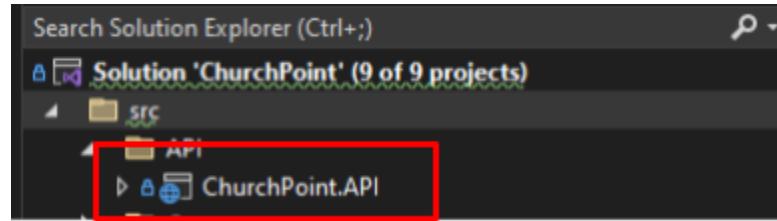
In order to deploy to sites other than **Azure**, make sure to release to the **IIS**. To do this follow these steps:

1. Navigate to the Solution Explorer and locate the **ChurchPoint.API** and **ChurchPoint.Blazor.Web** projects.

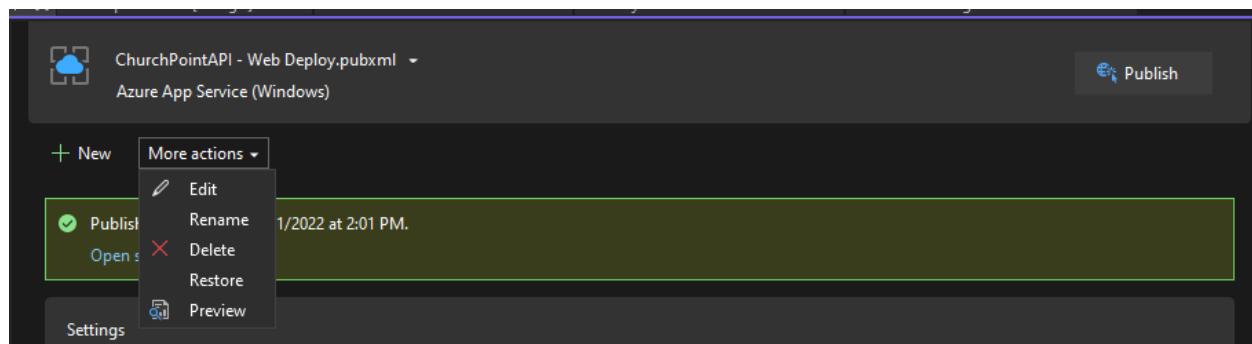


2. Then for each right-click them and hit publish.





3. After hitting publish use the drop down 'More Actions' and press Delete, to remove the publishing profile to Azure. Next, you will now click on publish and the target will be to Web Deploy to IIS.



4. For the rest of the process, make sure to follow the Microsoft Documentation in the following link to deploy to the IIS properly.

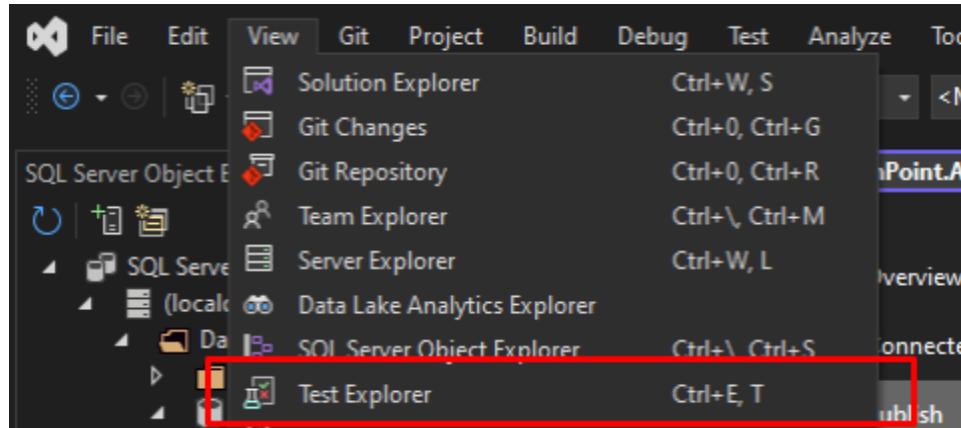
<https://docs.microsoft.com/en-us/visualstudio/deployment/tutorial-import-publish-settings-iis?view=vs-2022>



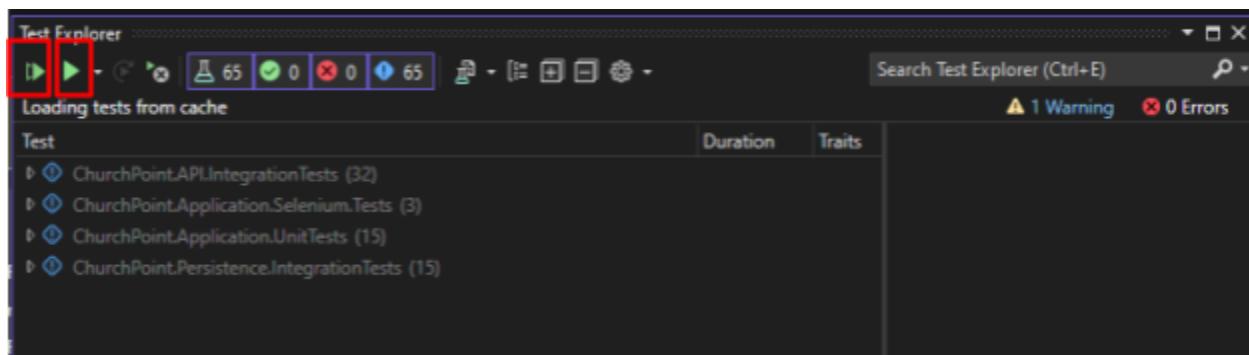
7 How to Run Automated Tests

In order to run the Automated Tests make sure to follow the steps for **Section 3 Configuring the Development Environment and Database**, otherwise the Automated Tests **will fail**. Then follow these steps:

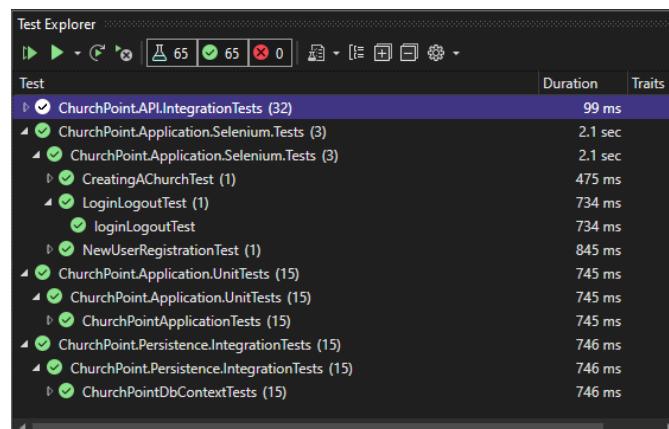
1. On Visual Studio 2022, go to the Top Menu Bar and click on View → Test Explorer



2. Then just click on the top left button to Run All Tests, or select tests to run and hit the button after the box red button.



3. The Tests should succeed like below

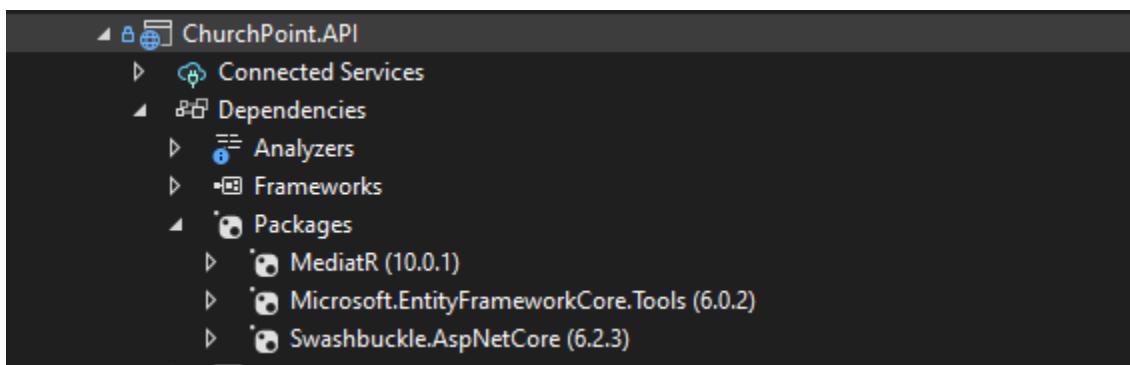
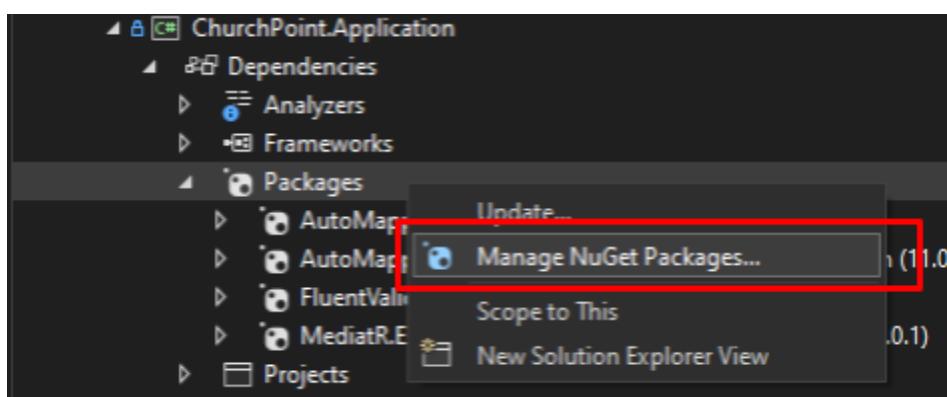


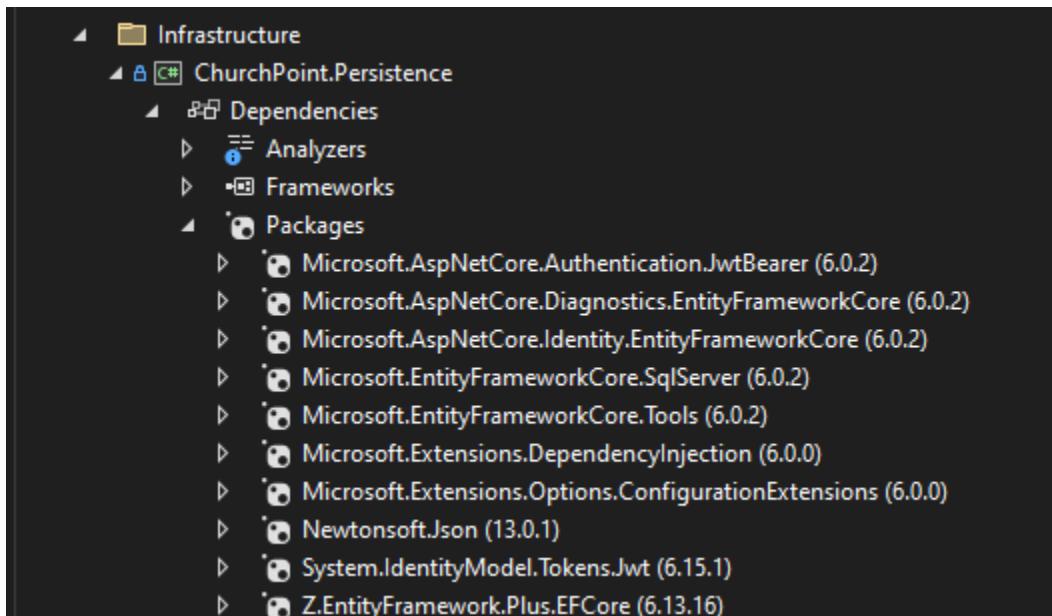
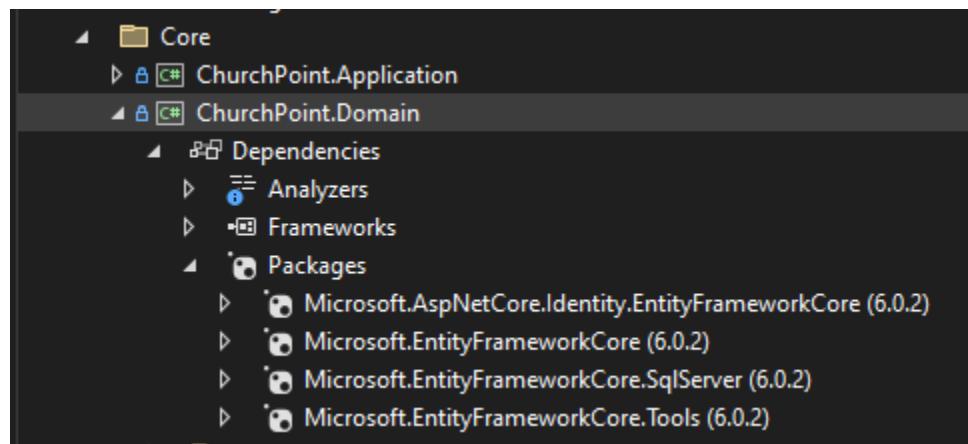
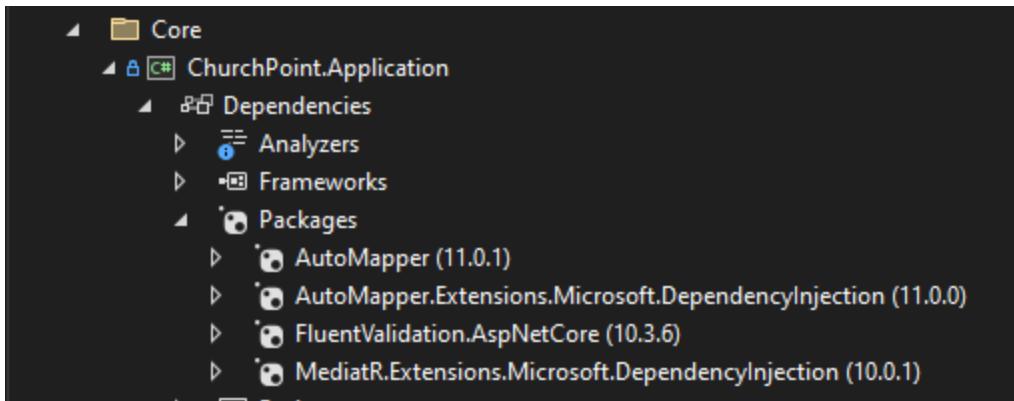
- To write more tests navigate to the Solution Explorer and look for the tests folder. Then follow the steps in **Section 4 How to Make A Code Change**

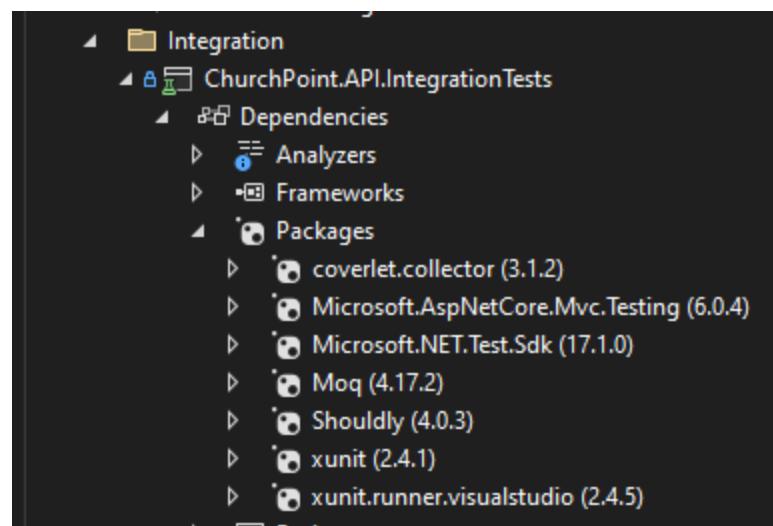
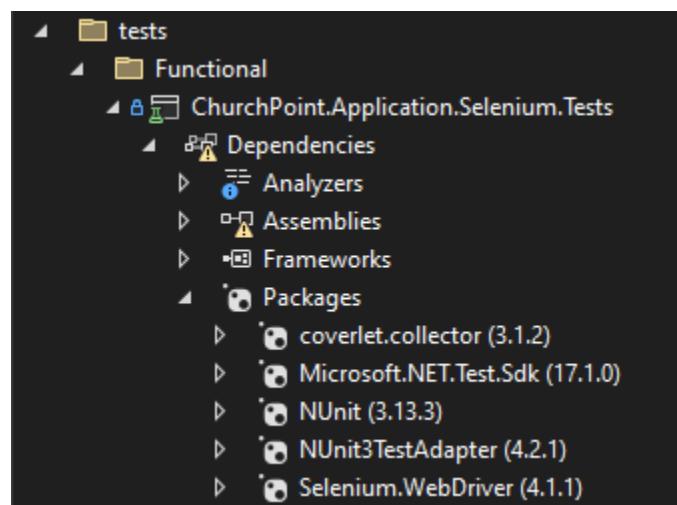
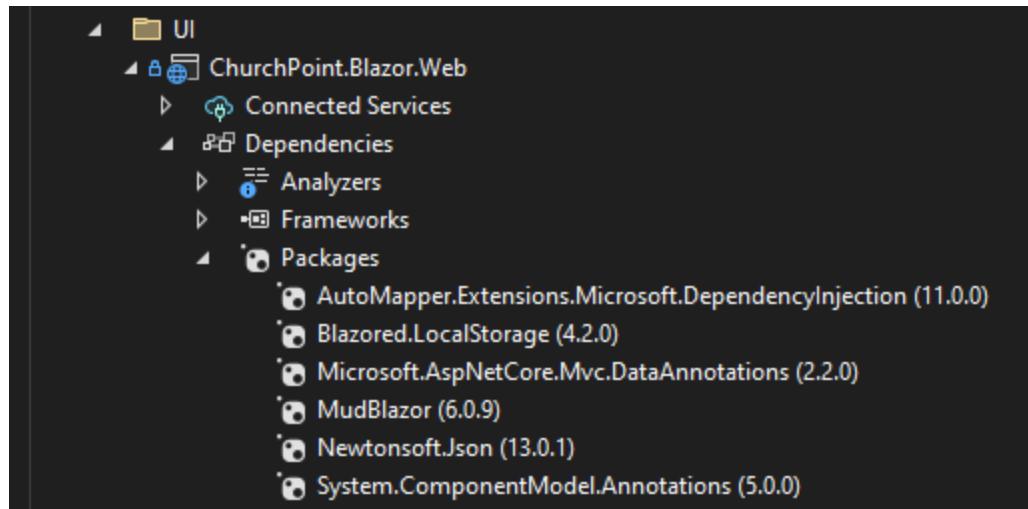
8 Checks For Runtime Assurance

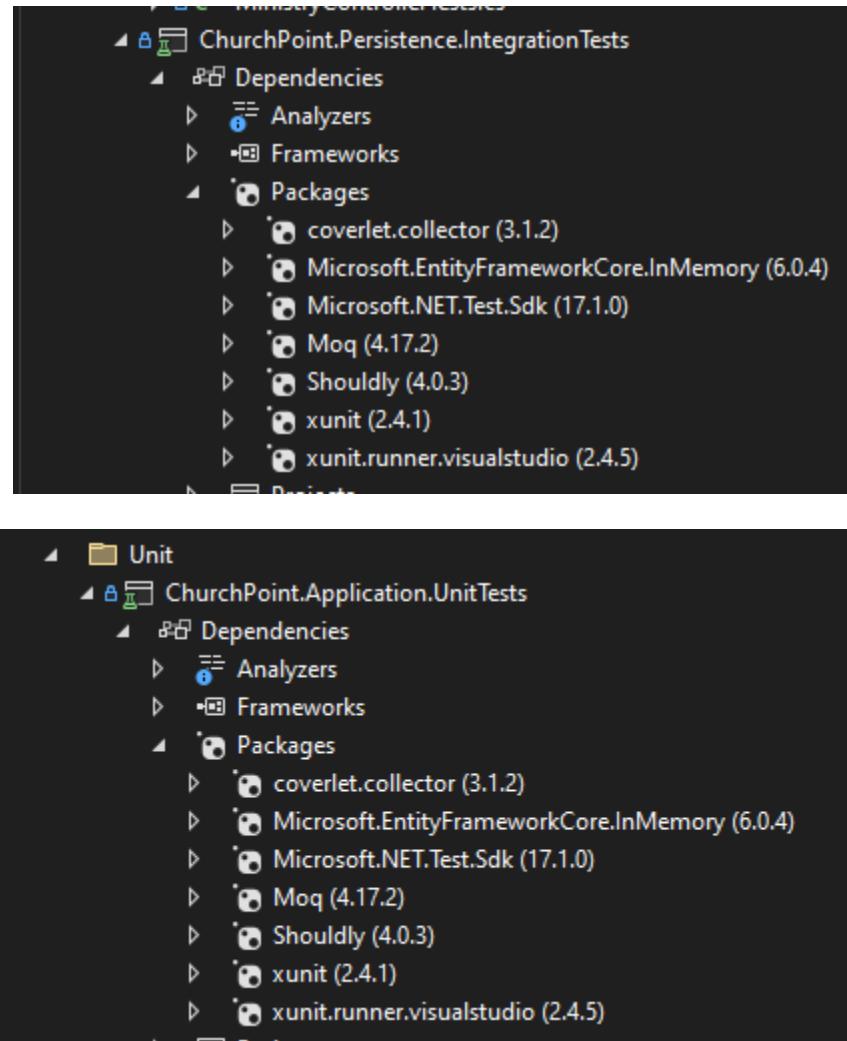
In order to check for runtime assurance make sure you have the following installed and follow the different sections as mentioned.

- .NET Core 6 Framework Installed
- Visual Studio 2022 Installed
- Appropriate NuGet packages per layer installed listed below: You would need to right-click the Project → Packages → Manage NuGet Packages
Then search for the packages listed below per Layer







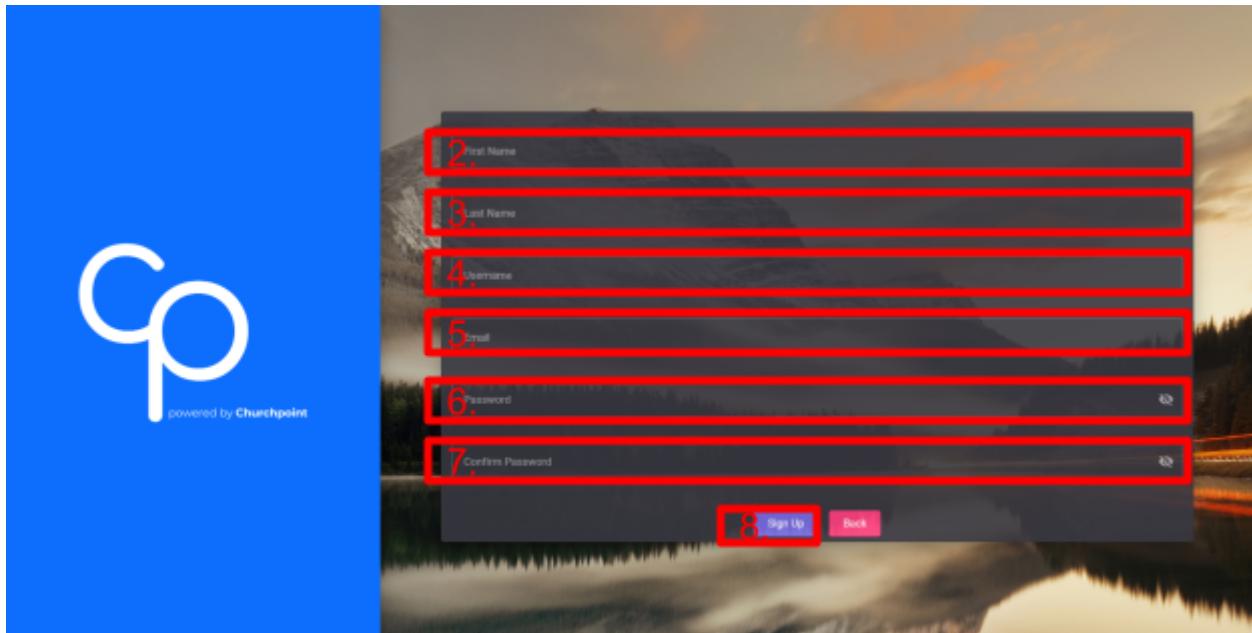
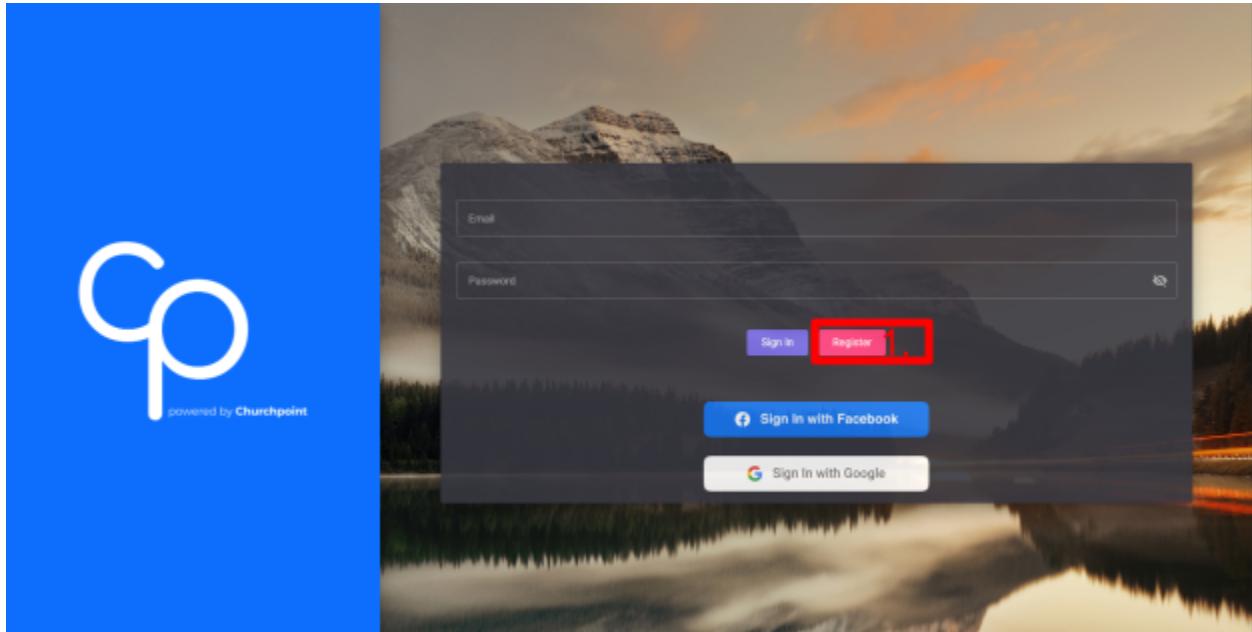


4. GitHub account created and integrated with Visual Studio 2022
NOTE for Steps 1-3 Refer to **Section 3 Configuring the Development Environment
5. Run all Tests found in **Section 7 How to Run Automated Tests**
6. If there are still problems that persist regarding anything within the project please contact the Project Manager at, pmapanao@csus.edu for more assistance.

9 ChurchPoint Front-End Web Client

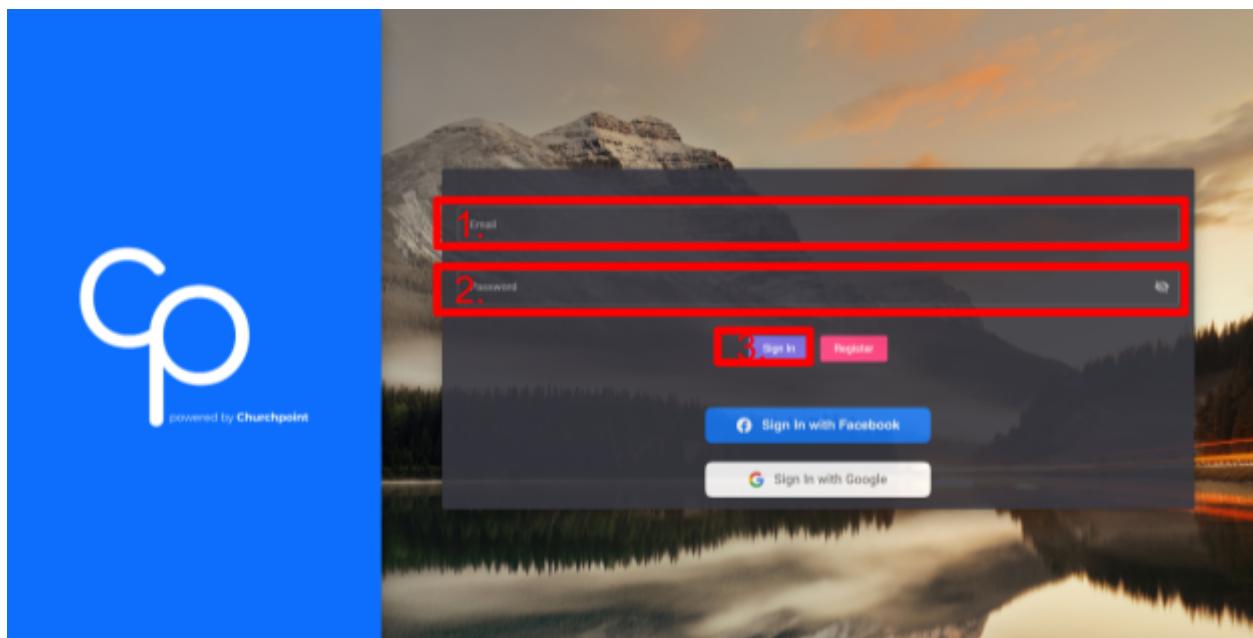
9.1 User Features

9.1.1 Registering a New Application User



Description	<p>To Register a new user, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Click Register on the Login Page 2. Enter First Name 3. Enter Last Name 4. Enter Username 5. Enter Email 6. Enter Password 7. Confirm Password 8. Click Sign Up
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web → Pages → Main Page → LoginPage</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Login.razor 2. LoginBase.cs <p>If other problems persist, check the following functions in the API section</p> <ol style="list-style-type: none"> 1. 10.2.1 Application User Authentication Register

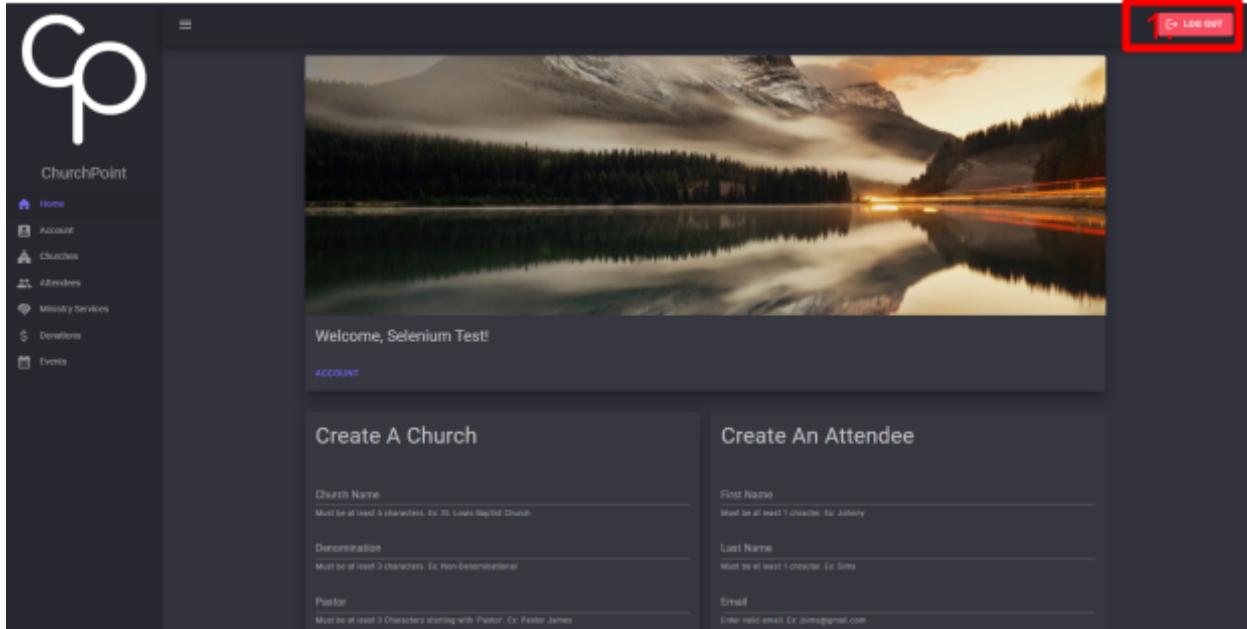
9.1.2 Logging in with an Existing Application User



Description	<p>To log in with an existing user, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Input the email that the user signed up with 2. Input the password of the user 3. Click Sign In
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web → Pages → Main Page → LoginPage</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Login.razor 2. LoginBase.cs <p>If other problems persist, check the following functions in the API section</p> <ol style="list-style-type: none"> 1. 10.2.1 Application User Authentication Authenticate 2. 10.2.3 Authorize



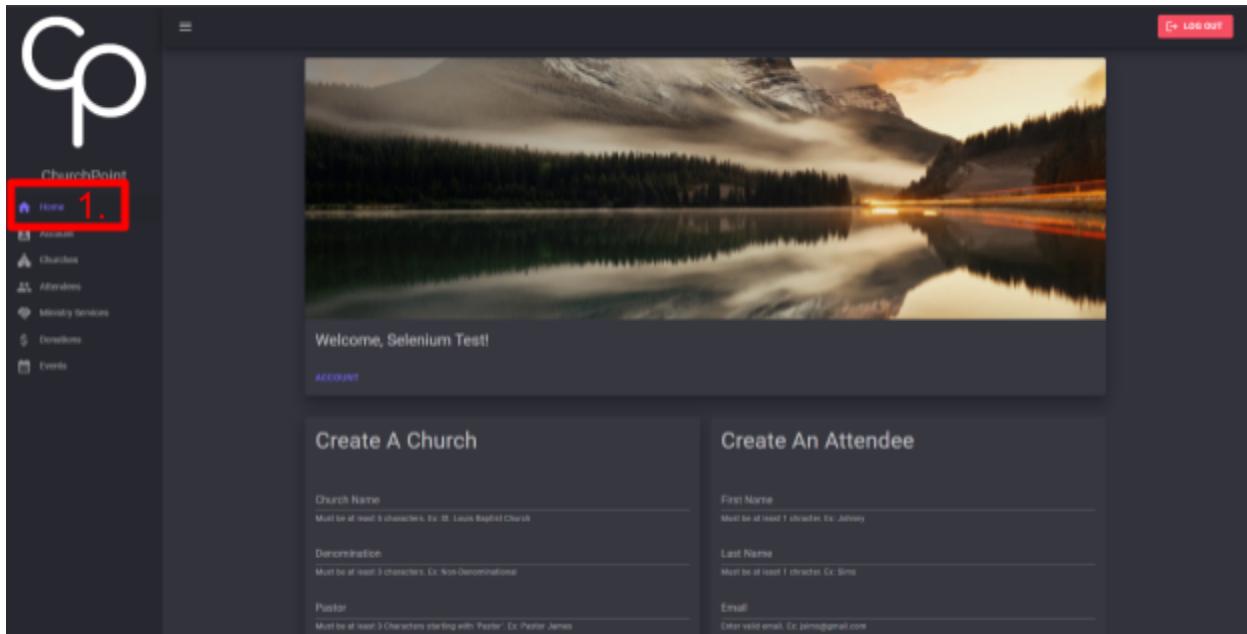
9.1.3 Logging out as an Application User



Description	To log out as an application user, simply look for the logout button on the top right-hand side of the home page.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Main Page</i> → <i>HomePage</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i> → <i>Base</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i> Then, check the following files: <ol style="list-style-type: none">1. Home.razor2. HomeBase.cs3. MainLayout.cs4. NavMenu.razor5. MainLayoutBase.cs If other problems persist, check the following functions pertaining to the following sections in the API section : <ol style="list-style-type: none">1. 10.2 API Authorize Access w/ Bearer Token2. 10.3 Application User API Functions3. 10.5 Churches API Functions4. 10.4 Attendees API Functions

9.2 Basic Navigation

9.2.1 Accessing Home Page



Description	To access the home page there will be a pop-out menu to the left of the website, click on the home tab and it will always direct the user back to the home page.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Main Page</i> → <i>HomePage</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i> → <i>Base</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none">1. Home.razor2. HomeBase.cs3. MainLayout.cs4. NavMenu.razor5. MainLayoutBase.cs <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none">1. 10.2 API Authorize Access w/ Bearer Token2. 10.3 Application User API Functions3. 10.5 Churches API Functions4. 10.4 Attendees API Functions

9.2.2 Accessing Account Page

The image consists of two screenshots of the ChurchPoint software interface.

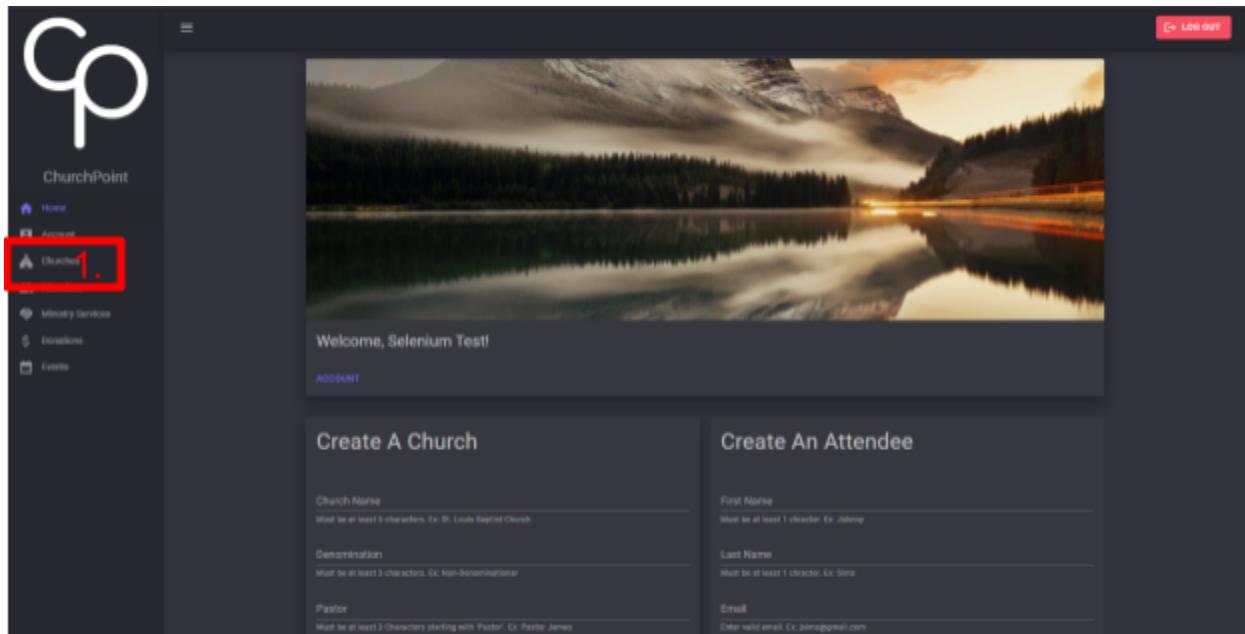
Top Screenshot: This shows the main dashboard of the ChurchPoint application. On the left is a sidebar with the ChurchPoint logo and navigation links: Home, Account (which is highlighted with a red box and labeled '1.'), Churches, Attendees, Ministry Services, Donations, and Events. The main content area features a scenic mountain lake background with the text "Welcome, Selenium Test!". Below this is a section titled "ACCOUNT" containing two forms: "Create A Church" and "Create An Attendee".

Bottom Screenshot: This shows the "Account Details" page. It has a red box around the form fields, which are identical to those in the top screenshot. The fields include "Email" (selenium@gmail.com), "Username" (seleniumtest), "First Name" (Selenium), and "Last Name" (Test). At the bottom of the form is a blue "SUMMIT CHANGES" button.



Description	To Access the Account Page, there will be a side menu on the left-hand side of the website, click on the account and it will automatically direct the user to the account page like in the second screenshot.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web → Pages → Main Page → HomePage</i> <i>ChurchPoint.Blazor.Web → Pages → Shared → Base</i> <i>ChurchPoint.Blazor.Web → Pages → Shared</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Home.razor 2. HomeBase.cs 3. MainLayout.cs 4. NavMenu.razor 5. MainLayoutBase.cs <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.2 API Authorize Access w/ Bearer Token 2. 10.3 Application User API Functions 3. 10.5 Churches API Functions 4. 10.4 Attendees API Functions

9.2.3 Accessing Churches Page



Churches

Church ID	Church Name	Denomination	Address	Person	Phone Number
a7bdc7b-d9c8-4710-8b53-71ef8be1e9	Celebrate Church	Episcopal	8834 Stark Ave Youngstown, Florida, 32495	John McKinna	408-976-9313
125be556-7325-4a6a-8083-40d178e33379	Bless Hill Refuge Church	African Methodist Episcopal	30 Dingwell St Level, Massachusetts, 01881	Alexander Brown	215-234-7228
8ba30c1-e2ee-4188-bb0e-fa18cf7f21212	Discovored Church	Orthodox	12380 FM 1725 Rd Cleveland, Texas, 77228	Iyler Gonzales	203-396-3198

Views per page: 10 < 1 of 3 >

Description	To Access the Churches Page, there will be a side menu on the left-hand side of the website, click on the Churches tab and it will automatically direct the user to the Churches page like in the second screenshot.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Main Page</i> → <i>HomePage</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i> → <i>Base</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Home.razor 2. HomeBase.cs 3. MainLayout.cs 4. NavMenu.razor 5. MainLayoutBase.cs <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.2 API Authorize Access w/ Bearer Token 2. 10.3 Application User API Functions 3. 10.5 Churches API Function 4. 10.4 Attendees API Functions



9.2.4 Accessing Attendees Page

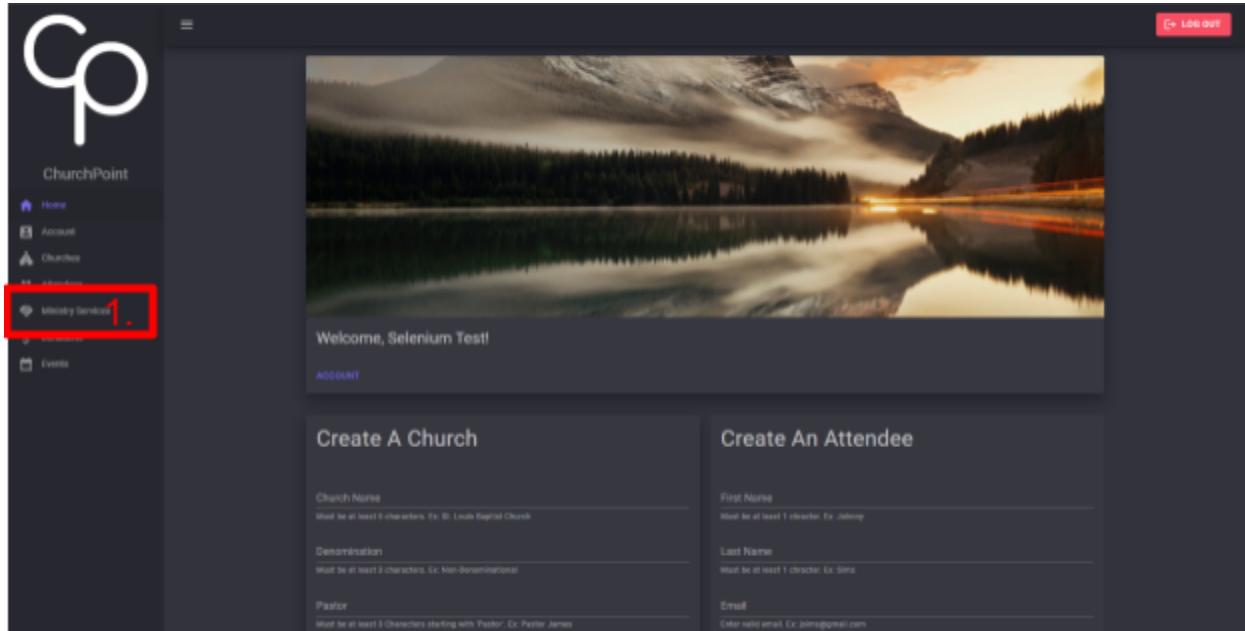
The screenshot shows the ChurchPoint software interface. On the left, a sidebar menu lists various options: Home, Account, Churches, **Attendees**, Ministry Services, Donations, and Events. The 'Attendees' option is highlighted with a red box and a large number '1.' above it. The main content area features a large banner image of a lake at sunset. Below the banner, a welcome message reads 'Welcome, Selenium Test!' and an account section shows 'ACCOUNT'. Two forms are displayed: 'Create A Church' on the left and 'Create An Attendee' on the right. The 'Create An Attendee' form includes fields for First Name, Last Name, and Email. At the bottom of the main content area, a table titled 'Attendees' lists several test attendees with columns for Attendee ID, First Name, Last Name, Phone Number, Address, and Email. A red box highlights this table, with a large number '2.' to its left. The top right corner of the screen has a 'LOG OUT' button.

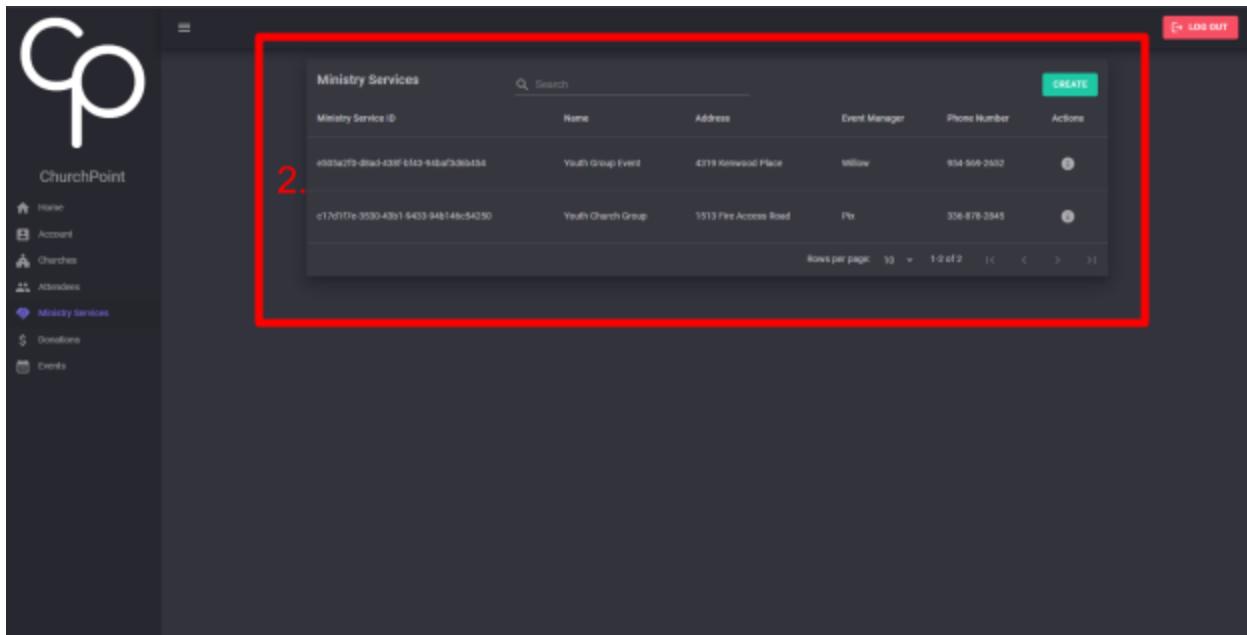
Attendee ID	First Name	Last Name	Phone Number	Address	Email
5a1fe8d-c753-4ba9-8111-7d20cf18cd32	Nicholas	Noel	423-617-7141	31 Park Lane Bryson City, North Carolina, 28713	nirayuhina@ecallen.com
0c2196ab-fa78-4769-a490-1fc5fca3d39	Test	Account	123-456-9845	789 Test Attendee Address 6 City State County	attendee@email.com
0a1bbdd-44f1-41ba-b131-bbd5759d52	Christopher	Morrise	882-880-2737	119 S Cedar St. Makonie, Illinois, 62059	Boaded@gallery.com
0c2196ab-fa78-4769-a490-1fc5fca3d39	Test	Account	123-456-9845	789 Test Attendee Address 6 City State County	attendee@email.com
dc0cc527-4329-4d1f-acf5-488ca810821	Hansley	Peterson	582-499-9222	Po Box 1616 Gambier, Ohio(34), 43022	militarygiberance.com
929b34c3-5532-44be-8048-c0462bd1d39	Aline	West	214-981-9277	281 Ed Hopkins St Dallas, Texas(TX), 77314	AlineWest@gmail.com



Description	To Access the Attendees Page, there will be a side menu on the left-hand side of the website, click on the Attendees tab and it will automatically direct the user to the Attendees page like in the second screenshot.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web → Pages → Main Page → HomePage</i> <i>ChurchPoint.Blazor.Web → Pages → Shared → Base</i> <i>ChurchPoint.Blazor.Web → Pages → Shared</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Home.razor 2. HomeBase.cs 3. MainLayout.cs 4. NavMenu.razor 5. MainLayoutBase.cs <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.2 API Authorize Access w/ Bearer Token 2. 10.3 Application User API Functions 3. 10.5 Churches API Function 4. 10.4 Attendees API Functions

9.2.5 Accessing Ministry Services Page





Description	To Access the Ministry Service Page, there will be a side menu on the left-hand side of the website, click on the Ministry Service tab and it will automatically direct the user to the Ministry Service page like in the second screenshot.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Main Page</i> → <i>HomePage</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i> → <i>Base</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Home.razor</i> 2. <i>HomeBase.cs</i> 3. <i>MainLayout.cs</i> 4. <i>NavMenu.razor</i> 5. <i>MainLayoutBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.2 API Authorize Access w/ Bearer Token 2. 10.3 Application User API Functions 3. 10.5 Churches API Function 4. 10.4 Attendees API Functions

9.2.6 Accessing Donations Page

The screenshot shows the ChurchPoint software interface. On the left is a sidebar with navigation links: Home, Account, Churches, Attendees, Ministry Services, Donations (which is highlighted with a red box and labeled '1.'), and Events. The main content area features a large landscape image of a lake at sunset. Below the image, a banner says "Welcome, Selenium Test!". Underneath the banner are two forms: "Create A Church" on the left and "Create An Attendee" on the right. Both forms have fields for First Name, Last Name, and Email, with validation messages below them.

The screenshot shows the ChurchPoint software interface. The sidebar on the left includes the "Donations" link, which is highlighted with a red box and labeled '2.'. The main content area displays a table titled "Donations". The table has columns: Donation ID, Church Name, Title, Date, Amount, Type, and Actions. One row is visible in the table, showing a donation from "Celebrate Church" dated "5/5/2022 9:32:53 AM" for "\$5" via "Cash". There are also buttons for "CREATE" and "SEARCH". At the bottom of the table, there are pagination controls and a "Rows per page" dropdown set to 10.

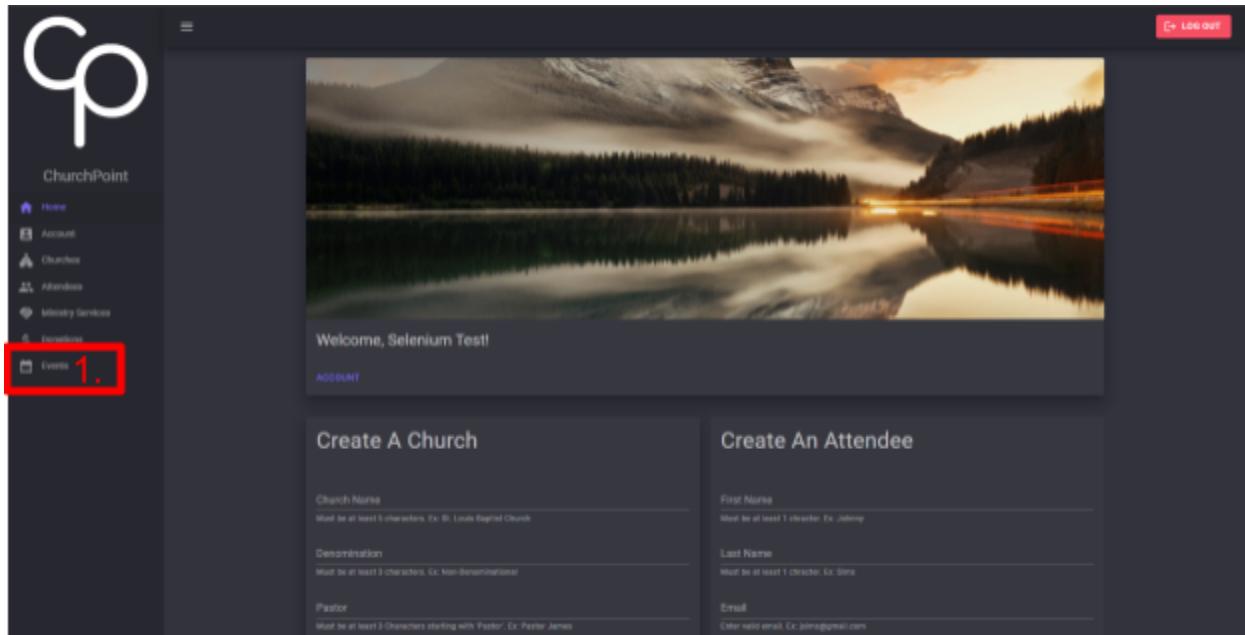
Donation ID	Church Name	Title	Date	Amount	Type	Actions
c09efac7-90e2-46a7-82d9-e538b31d629	Celebrate Church	Donation	5/5/2022 9:32:53 AM - \$7.00	\$5	Cash	



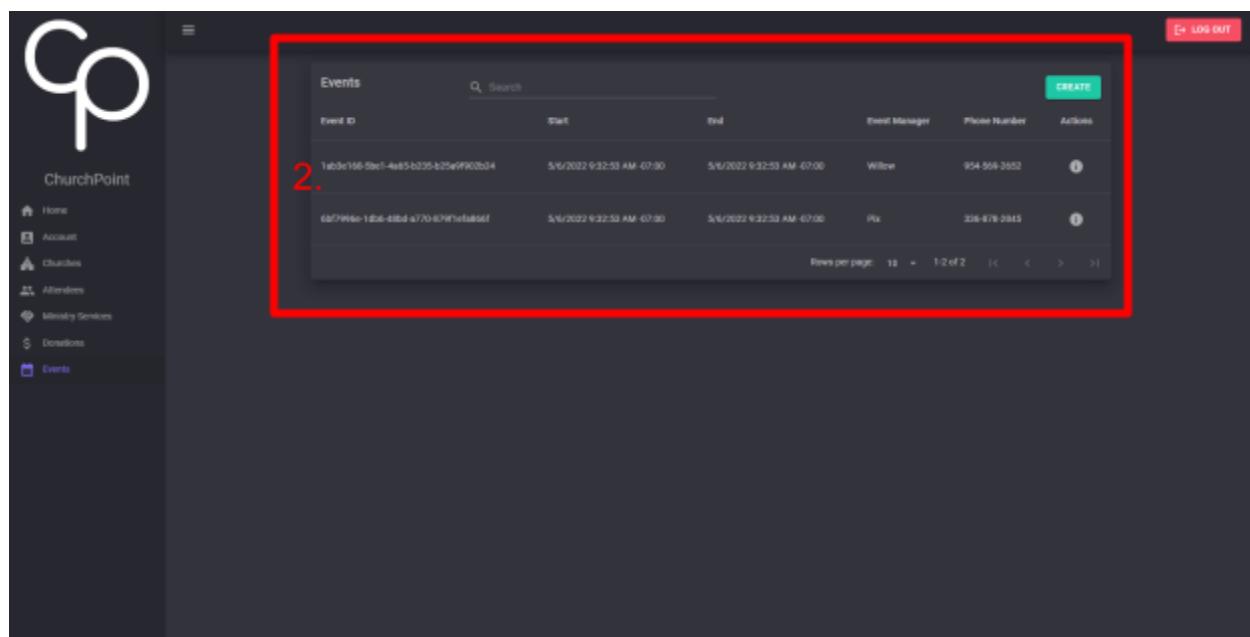
Description	To Access the Donations Page, there will be a side menu on the left-hand side of the website, click on the Donations tab and it will automatically direct the user to the Donations page like in the second screenshot.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web → Pages → Main Page → HomePage</i> <i>ChurchPoint.Blazor.Web → Pages → Shared → Base</i> <i>ChurchPoint.Blazor.Web → Pages → Shared</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Home.razor 2. HomeBase.cs 3. MainLayout.cs 4. NavMenu.razor 5. MainLayoutBase.cs <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.2 API Authorize Access w/ Bearer Token 2. 10.3 Application User API Functions 3. 10.5 Churches API Function 4. 10.4 Attendees API Functions



9.2.7 Accessing Events Page



The screenshot shows the ChurchPoint software interface. On the left, there is a dark sidebar with the ChurchPoint logo at the top. Below the logo, the sidebar has several menu items: Home, Account, Churches, Attendees, Ministry Services, Donations, and Events. The 'Events' item is highlighted with a red box and a number '1.' next to it. The main content area has a large banner image of a lake and mountains. Below the banner, the text 'Welcome, Selenium Test!' is displayed. Underneath this, there are two forms: 'Create A Church' on the left and 'Create An Attendee' on the right. Both forms have input fields for Church Name, Denomination, Pastor, First Name, Last Name, and Email.



This screenshot shows the same ChurchPoint interface as the previous one, but the main content area now displays a list of events. A red box highlights this list, which is titled 'Events'. The list includes two entries:

Event ID	Start	End	Event Manager	Phone Number	Actions
Tab03c168-5be1-4e85-b235-b25e99903034	5/6/2022 9:32:53 AM -07:00	5/6/2022 9:32:53 AM -07:00	Wilma	954-554-2852	
6d7D946c-18d6-41bd-a770-679f1efab86f	5/6/2022 9:32:53 AM -07:00	5/6/2022 9:32:53 AM -07:00	Pix	336-878-2845	

At the bottom of the event list, there is a pagination control labeled 'Pages per page: 10 < < > >>'.

Description	To Access the Events Page, there will be a side menu on the left-hand side of the website, click on the Events tab and it will automatically direct the user to the Events page like in the second screenshot.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web → Pages → Main Page → HomePage</i>

ChurchPoint.Blazor.Web → Pages → Shared → Base
ChurchPoint.Blazor.Web → Pages → Shared

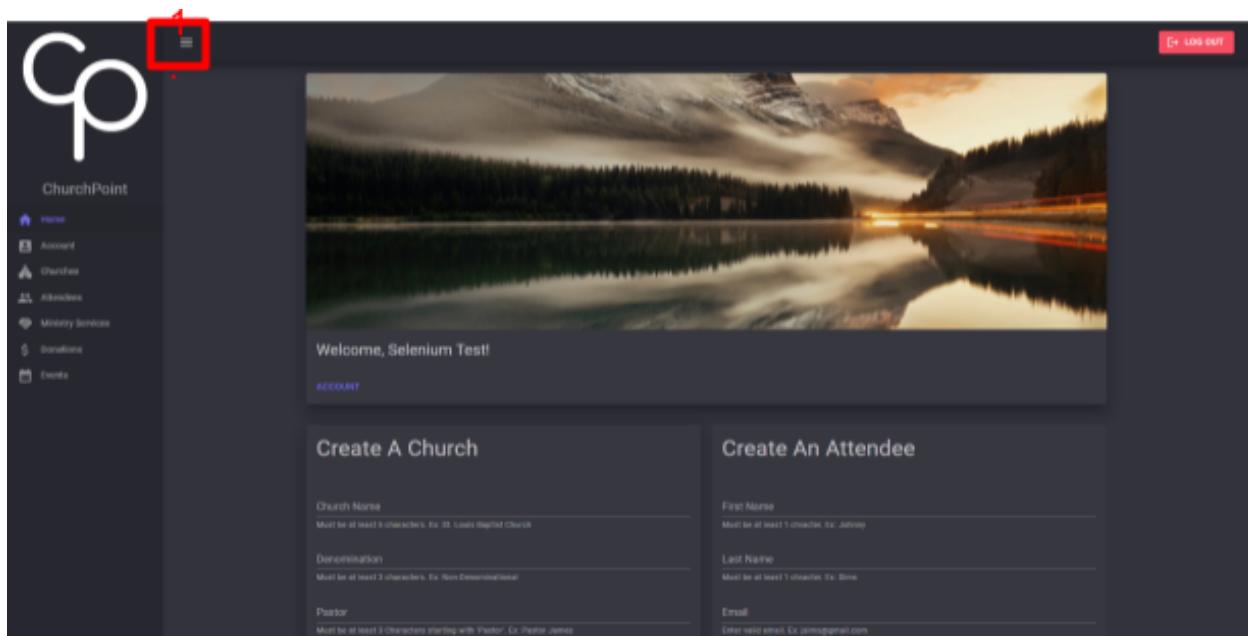
Then, check the following files:

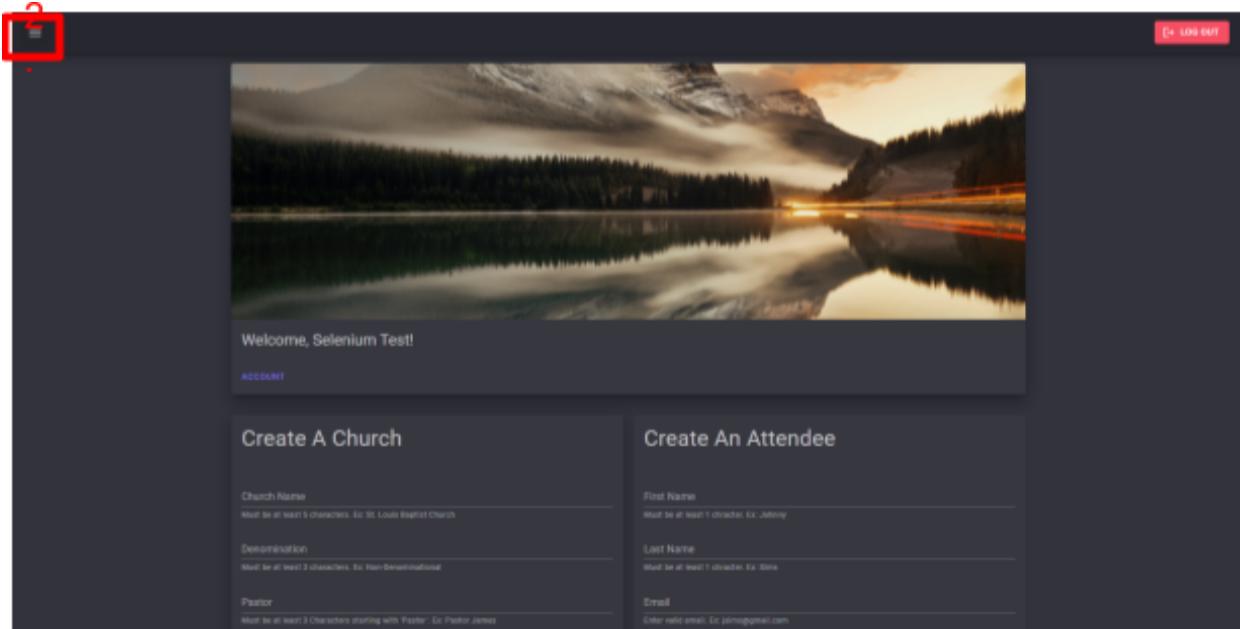
1. Home.razor
2. HomeBase.cs
3. MainLayout.cs
4. NavMenu.razor
5. MainLayoutBase.cs

If other problems persist, check the following functions pertaining to the following sections in the **API section**:

1. **10.2 API Authorize Access w/ Bearer Token**
2. **10.3 Application User API Functions**
3. **10.5 Churches API Function**
4. **10.4 Attendees API Functions**

9.2.8 Hiding/Showing the Navigation Bar





Description	To hide/show the navigation bar, there will be a hamburger icon off to the top left of the website. Picture 1 shows the location of the icon when the navigation bar is open and picture 2 shows the location of the icon when the navigation bar is closed.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Main Page</i> → <i>HomePage</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i> → <i>Base</i> <i>ChurchPoint.Blazor.Web</i> → <i>Pages</i> → <i>Shared</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Home.razor 2. HomeBase.cs 3. MainLayout.cs 4. NavMenu.razor 5. MainLayoutBase.cs <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.2 API Authorize Access w/ Bearer Token 2. 10.3 Application User API Functions 3. 10.5 Churches API Function 4. 10.4 Attendees API Functions

9.3 Church Features

9.3.1 Viewing Churches within the Database

The screenshot shows the ChurchPoint software interface. On the left is a sidebar with the ChurchPoint logo and navigation links: Home, Account, Churches (which is selected and highlighted in blue), Addresses, Ministry Services, Donations, and Events. The main content area has a title 'Churches' and a search bar. Below is a table with columns: Church ID, Church Name, Denomination, Address, Pastor, and Phone Number. Three rows of church data are listed. At the top right of the table are 'ADD' and 'CREATE' buttons, with 'ADD' being highlighted with a red box. At the bottom right are pagination controls and a 'Rows per page' dropdown.

Church ID	Church Name	Denomination	Address	Pastor	Phone Number
a7fbec70-d5e8-47fe-8f53-711ff8a918	Celestial Church	Episcopal	6834 Stark Ave Youngstown, Florida, 32488	John McNamee	406-675-6113
1205a856-7309-4a8a-8890-43b178cd1325	Bless Hill Refuge Church	African Methodist Episcopal	38 Bragwell St Lowell, Massachusetts, 01853	Alexander Brown	319-334-7028
#0ca3d1-a3ae-4188-b6de-fa19cf1f510	Discovered Church	Orthodox	12380 Fm 1729 Rd Cleveland, Texas, 77528	Hector Gonzales	281-399-2199

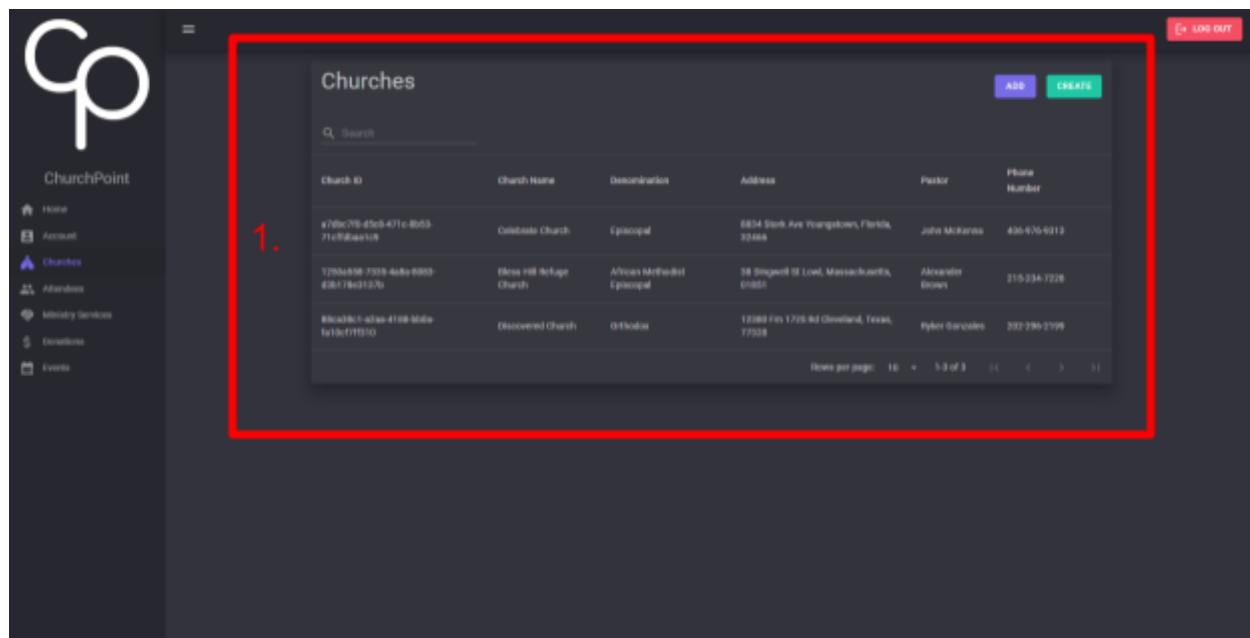
The screenshot shows a modal dialog titled 'Select Churches' over the ChurchPoint interface. The sidebar on the left is identical to the previous screenshot. The modal contains a search bar and a table with columns: Church ID, Church Name, and Pastor. It lists six churches: Alist Church, Entrance Gospel Church, Resonate Restoration Church, Test Church, The River, and a 'Select All' option. At the bottom right of the modal is a 'CONFIRM' button. A large red box surrounds the entire modal area, and the number '2.' is placed to the left of the modal's title bar.

Church ID	Church Name	Pastor
a7fbec70-d5e8-47fe-8f53-711ff8a918	Alist Church	Alan White
77b63c10-e472-4084-ae05-10eb7af3aef0	Entrance Gospel Church	Sloane Torres
c0556479-ebab-45d0-99c1-4e21eef11e68	Resonate Restoration Church	Violet Powley
18124ef0-ee05-4bea-b3a5-103399bd1d1c	Test Church	Pastor TestName
a10ea4e5-1060-4149-ae05-5a3850322999	The River	Caroline Hassley
<input type="checkbox"/> Select All		



Description	To view the churches within the database, navigate to the Churches Page. 1. Click the add button on the right-hand side 2. A list of the churches within the database will appear
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i> Then, check the following files: 1. <i>Church.razor</i> 2. <i>ChurchBase.cs</i> 3. <i>ChurchDetail.razor</i> 4. <i>ChurchDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : 1. 10.5 Churches API Functions

9.3.2 Viewing Churches assigned to Application User



The screenshot shows the ChurchPoint software interface. On the left is a sidebar with the ChurchPoint logo and navigation links: Home, Account, Churches, Addresses, Ministry Services, Donations, and Events. The 'Churches' link is highlighted. The main area is titled 'Churches' and contains a table with three rows of church data. A red box highlights the first row. The table columns are: Church ID, Church Name, Denomination, Address, Pastor, and Phone Number.

Church ID	Church Name	Denomination	Address	Pastor	Phone Number
870bc79-45eb-471e-8d53-71c0ff8e9f8	Celestial Church	Episcopal	6834 Stark Ave Youngstown, Florida, 33468	John McAnena	436-676-9113
1296a858-7339-4d8a-8093-680178a01329	Bless Hill Refuge Church	African Methodist Episcopal	18 Bingley St Lowell, Massachusetts, 01801	Alexander Brown	215-334-7228
88ca0f1-47aa-4188-b0de-fa193cf1f510	Discovered Church	Orthodox	12380 Fies 1725 Rd Cleveland, Texas, 77528	Ryley Gonzales	282-396-2199



Description	To view the churches assigned to the application user, navigate to the churches page, A list of the churches assigned to the application user will appear
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i></p> <p><i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Church.razor</i> 2. <i>ChurchBase.cs</i> 3. <i>ChurchDetail.razor</i> 4. <i>ChurchDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <p>1. 10.5 Churches API Functions</p>

9.3.3 Viewing Church Detail Page

The screenshot shows the ChurchPoint software interface with a dark theme. On the left, a sidebar menu includes links for Home, Announcements, Churches (which is selected and highlighted in blue), Attendance, Ministry Services, Donations, and Events. The main content area is titled 'Churches' and displays a list of three churches. Each church entry includes fields for Church ID, Church Name, Denomination, Address, Pastor, and Phone Number. The first church listed is 'Celebrate Church' (Episcopal). The second is 'Bless Hill Refuge Church' (African Methodist Episcopal). The third is 'Discovered Church' (Orthodox). At the bottom of the list, there is a pagination control showing '1 of 3' and arrows for navigation. In the top right corner of the main area, there are 'Add' and 'CREATE' buttons. The 'Edit To Detail Page' button for the first church is highlighted with a red box.



The screenshot shows the 'Church Details' page of the ChurchPoint Blazor application. The page has a dark theme with a sidebar on the left containing navigation links: Home, Account, Churches, Attendees, Ministry Services, Operations, and Events. The main content area is titled 'Church Details' and contains the following fields:

- SUID:** a7b0c7f0-d5c8-4710-8953-71cf5baa1c9 (Global Unique Identifier)
- Church Name:** Celebrate Church (Must be at least 5 characters, e.g. St. Louis Baptist Church)
- Denomination:** Episcopal (Must be at least 3 characters, e.g. Non-Denominational)
- Pastor:** John McKenna (Must be at least 3 characters starting with 'Pastor', e.g. Pastor James)
- Address:** 8824 Stark Ave Youngstown, Florida, 32466 (Must be a full US address, Example: 234 Grapene Rd, New City, State 78945)
- Phone Number:** 406-976-9313 (Must be a phone number with area code, e.g: 123-456-2342)

At the bottom of the form are two buttons: 'SUBMIT CHANGES' (blue) and 'RESET' (yellow) and 'DELETE' (red).

Description	<p>To access the Church Details Page:</p> <ol style="list-style-type: none"> 1. Navigate the Church Page 2. Click on the church to which details are needed, an information icon will appear 3. Click on the information icon and it will direct the user to the Church Details Page
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i></p> <p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Church.razor</i> 2. <i>ChurchBase.cs</i> 3. <i>ChurchDetail.razor</i> 4. <i>ChurchDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.5 Churches API Functions

9.3.4 Paging: Setting Amount of Churches to View Within Church List Menu

The screenshot shows the ChurchPoint software interface with the 'Churches' list page. The left sidebar includes links for Home, Account, Churches (which is selected), Attendance, Ministry Services, Devotions, and Events. The main content area displays a table with columns: Church ID, Church Name, Denomination, Address, Pastor, and Phone Number. Three rows of church data are listed. Below the table is a pagination control with a dropdown labeled 'Rows per page' containing options: 5, 10, 25, 50, and 100. The number '1.' is written in red inside a red box that surrounds the entire 'Rows per page' dropdown area.

Description	To set the number of churches to view within the Church List menu first navigate to the church page, within the church page there will be a Rows per page with a dropdown. Click the drop-down menu and the user can modify how many churches can be seen.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i></p> <p><i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Church.razor</i> 2. <i>ChurchBase.cs</i> 3. <i>ChurchDetail.razor</i> 4. <i>ChurchDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.5 Churches API Functions

9.3.5 Paging: Changing Between Pages Within the Church List Menu

Church ID	Church Name	Denomination	Address	Pastor	Phone Number
1620e0d8-ed5-40ea-bbb5-193096bf1f18	Test Church	Test Denomination for this Test Church	123 Test Church Addr 12345 City State Country	Pastor Test Name	123-456-7890
77503c30-e473-42b4-ac35-1be67a6a0d00	Embrace Gospel Church	Baptist General Conference	5675-A Utica Pky Meadow, New York, 11365	Silence Tones	517-754-2321
e858047e-e5d4-4563-9b04-4aefecf4fe00	Resolute Restoration Church	Anglican	3031 Wolf Lake New Albany, Indiana, 47150	Violet Poetry	317-659-7801
edfc1194-fab4-4149-bb95-6dd603fec538	Alive Church	Association of Free Lutheran Congregations	8440 Quail Run Ct Dallas, North Carolina, 28609	Alan White	334-931-6856
a20bc110-d5d8-4710-6b55-7fcdff8e1e9	Celebrate Church	Episcopal	8634 Stock Ave Youngstown, Florida, 32486	John Morrison	408-476-6313

Rows per page: 5 | < < > >> | 1 |

Description	To change between pages within the church list menu, there are some left and right arrows towards the bottom right of the church page. Click on the single arrow to move one page or on the arrow with a line to navigate to the last page.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i></p> <p><i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Church.razor</i> 2. <i>ChurchBase.cs</i> 3. <i>ChurchDetail.razor</i> 4. <i>ChurchDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.5 Churches API Functions



9.3.6 Searching for Churches Using the Church List Menu

Church ID	Church Name	Denomination	Address	Pastor	Phone Number
a7ebc7b-d3cb-471c-8553-71ef5feaf1e9	Odelete Church	Episcopal	8834 Stark Ave Youngstown, Florida, 37496	John McKinna	806-678-5313
129a556-7325-4a8a-8963-63b178e021379	Bless Hill Refuge Church	African Methodist Episcopal	36 Wellington St Lowell, Massachusetts, 01851	Alexander Brown	215-234-7228
88ea98cf-a3e8-4f68-b60e-fa3cf7fb210	Discovered Church	Orthodox	12390 Elm 1725 Rd Cleveland, Texas, 77228	Iyler Gonzales	203-396-2199

Description	To Search for churches using the church list menu, there will be a search bar towards the top left of the church page. Click on the search and input the details for the necessary search
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i> Then, check the following files: <ol style="list-style-type: none">1. <i>Church.razor</i>2. <i>ChurchBase.cs</i>3. <i>ChurchDetail.razor</i>4. <i>ChurchDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : <ol style="list-style-type: none">1. 10.5 Churches API Functions



9.3.7 Creating New Churches

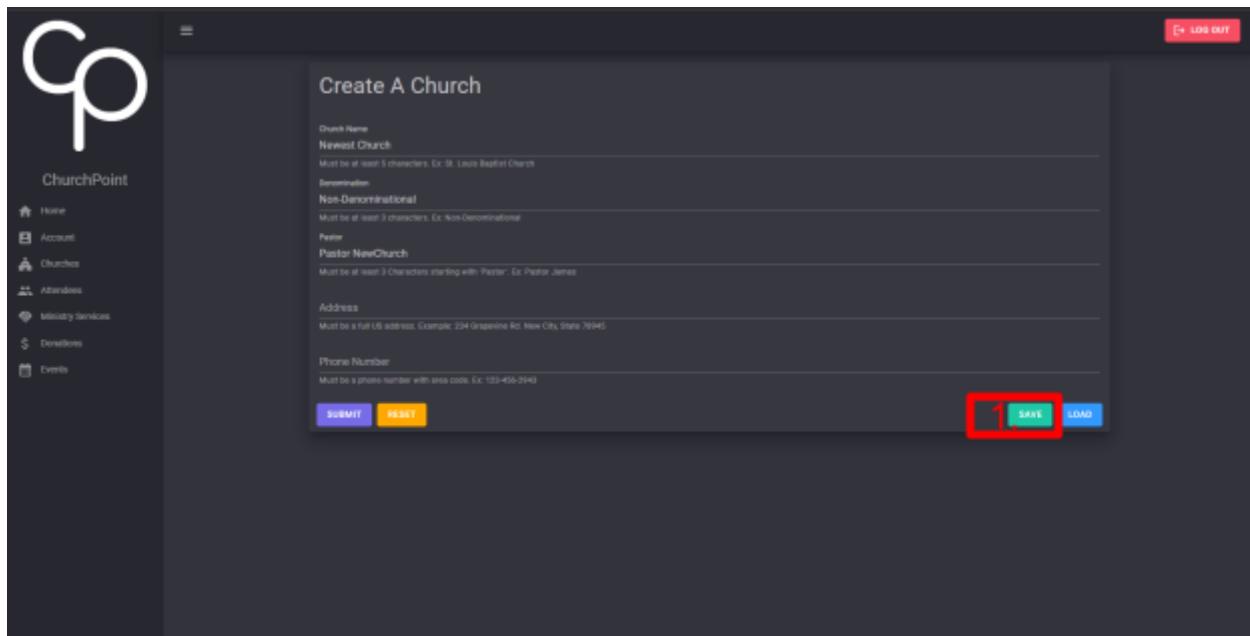
The screenshot shows the ChurchPoint software interface. On the left is a sidebar with the ChurchPoint logo and navigation links: Home, Account, Churches (which is selected and highlighted in blue), Attendees, Ministry Services, Donations, and Events. The main content area is titled 'Churches'. At the top right of this area are two buttons: 'ADD' and 'CREATE', with 'CREATE' being highlighted by a red box. Below this is a search bar labeled 'Search'. A table lists three churches: 'Celebrate Church' (Episcopal), 'Mesa Hill Refuge Church' (African Methodist Episcopal), and 'Discovered Church' (Orthodox). Each row includes columns for Church ID, Church Name, Denomination, Address, Pastor, and Phone Number. At the bottom of the table are buttons for 'Rows per page' (set to 10), '1 of 3', and navigation arrows.

The screenshot shows the 'Create A Church' form. The entire form is highlighted by a large red box. The number '2.' is written in red at the top left of the form area. The form fields include: 'Church Name' (with a note: 'Must be at least 5 characters. Ex: St. Louis Baptist Church'), 'Denomination' (with a note: 'Must be at least 3 characters. Ex: Non-Denominational'), 'Pastor' (with a note: 'Must be at least 3 characters starting with "Pastor". Ex: Pastor James'), 'Address' (with a note: 'Must be a full US address. Example: 234 Sappine Rd, New City, State 78945'), and 'Phone Number' (with a note: 'Must be a phone number with area code. Ex: 123-456-2345'). At the bottom are 'SUBMIT' and 'RESET' buttons on the left, and 'SAVE' and 'LOAD' buttons on the right. In the top right corner of the form area is a 'Cancel Out' button.



Description	<p>To create a church navigate to the Church page</p> <ol style="list-style-type: none"> 1. Within the Church page, there will be a create button on the top right, click on button 2. The user will be redirected to the create a church page
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i></p> <p><i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Church.razor</i> 2. <i>ChurchBase.cs</i> 3. <i>ChurchDetail.razor</i> 4. <i>ChurchDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.5 Churches API Functions

9.3.8 Temporarily Saving New Churches to be Added

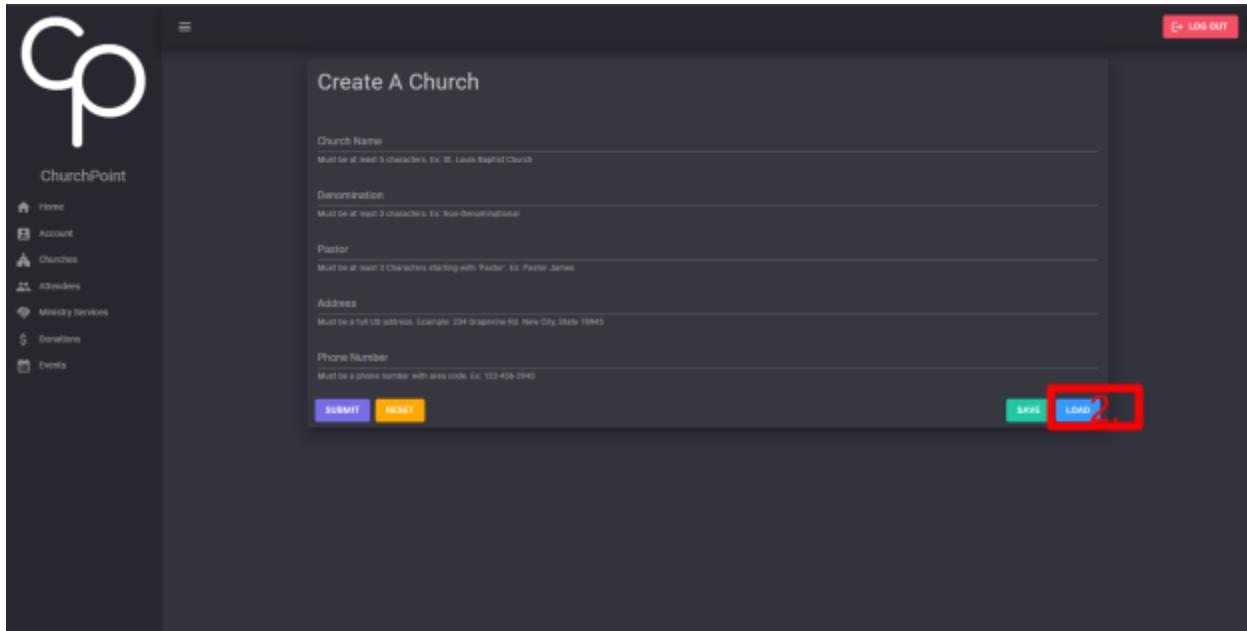


The screenshot shows the 'Create A Church' page from the ChurchPoint software. The page has a dark background with white text. On the left is a sidebar with the ChurchPoint logo and links to Home, Account, Churches, Addresses, Ministry Services, Donations, and Events. The main area is titled 'Create A Church'. It contains several input fields with validation messages:

- Church Name:** Newest Church (Must be at least 5 characters. Ex: St. Louis Baptist Church)
- Denomination:** Non-Denominational (Must be at least 3 characters. Ex: Non-Denominational)
- Pastor:** Pastor NewChurch (Must be at least 2 characters starting with 'Pastor'. Ex: Pastor James)
- Address:** (Must be a full U.S address. Example: 234 Grapine Rd, New City, State 78945)
- Phone Number:** (Must be a phone number with area code. Ex: 123-456-2345)

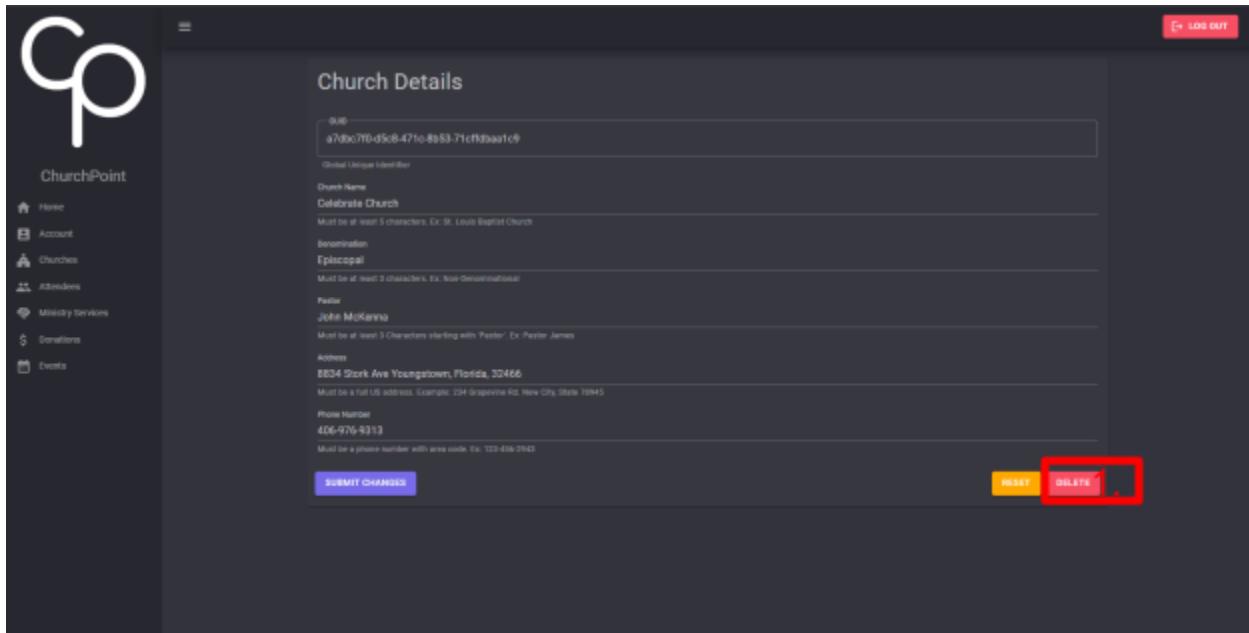
At the bottom, there are three buttons: 'SUBMIT' (disabled), 'RESET', and a 'SAVE' button which is highlighted with a red box.





Description	To temporarily save new churches to be added, navigate to the create a church page, please refer to the 2.3.7, within the create a church page there will be a save button once clicked the information will be saved. the user will have that information saved and would have to click the load button if for any reason there is a need to navigate away or accidentally erasing data.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Church.razor</i> 2. <i>ChurchBase.cs</i> 3. <i>ChurchDetail.razor</i> 4. <i>ChurchDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.5 Churches API Functions

9.3.9 Removing Churches From an Application User



Description	Navigate to the Church Details page, refer to 2.3.3, within the church details page there will be a delete button on the lower right-hand side of the website where the user can delete the church from their list if needed.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i> Then, check the following files: <ol style="list-style-type: none">1. <i>Church.razor</i>2. <i>ChurchBase.cs</i>3. <i>ChurchDetail.razor</i>4. <i>ChurchDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : <ol style="list-style-type: none">1. 10.5 Churches API Functions

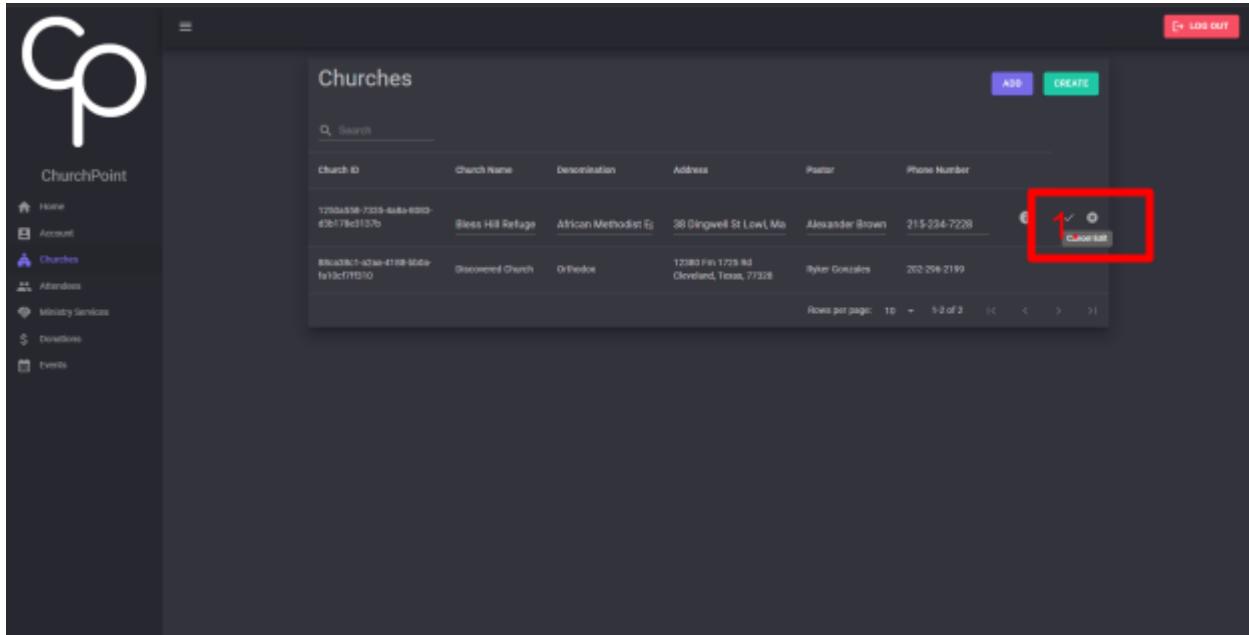
9.3.10 Quick Editing Using the Church List Menu

The screenshot shows the ChurchPoint software interface. On the left is a sidebar with the ChurchPoint logo and navigation links: Home, Account, Churches (which is selected and highlighted in blue), Attendees, Ministry Services, Donations, and Events. The main area is titled 'Churches' and contains a table with two rows of data. The first row is highlighted with a red box. The columns are labeled: Church ID, Church Name, Denomination, Address, Pastor, and Phone Number. The first row's data is: 1234567890-1234567890-1234567890-1234567890-1234567890-1234567890, Bless Hill Refuge, African Methodist E., 38 Dingwall St Lowell, Ma, Alexander Brown, 215-234-7228. The second row's data is: 0987654321-0987654321-0987654321-0987654321-0987654321-0987654321, Discovored Church, Orthodox, 12345 Elm 1725 Rd Cleveland, Texas, 77328, Ryker dianoles, 282-285-2196. At the top right of the main area are 'LOG OUT' and 'CREATE' buttons. Below the table are buttons for 'Rows per page' (10, 20, 50, 100) and navigation arrows.

Description	To quick edit using the church list menu, navigate to the church page, within the church page click on one of the churches and the user will be able to quickly edit the church name, denomination, address, pastor, and phone number
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i> Then, check the following files: <ol style="list-style-type: none">1. <i>Church.razor</i>2. <i>ChurchBase.cs</i>3. <i>ChurchDetail.razor</i>4. <i>ChurchDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : <ol style="list-style-type: none">1. 10.5 Churches API Functions



9.3.11 Canceling Quick Editing Using the Church List Menu



The screenshot shows the ChurchPoint software interface with the 'Churches' list page. The left sidebar includes links for Home, Account, Churches (which is selected), Attendees, Ministry Services, Donations, and Events. The main area displays a table with columns: Church ID, Church Name, Denomination, Address, Pastor, and Phone Number. Two rows of data are visible:

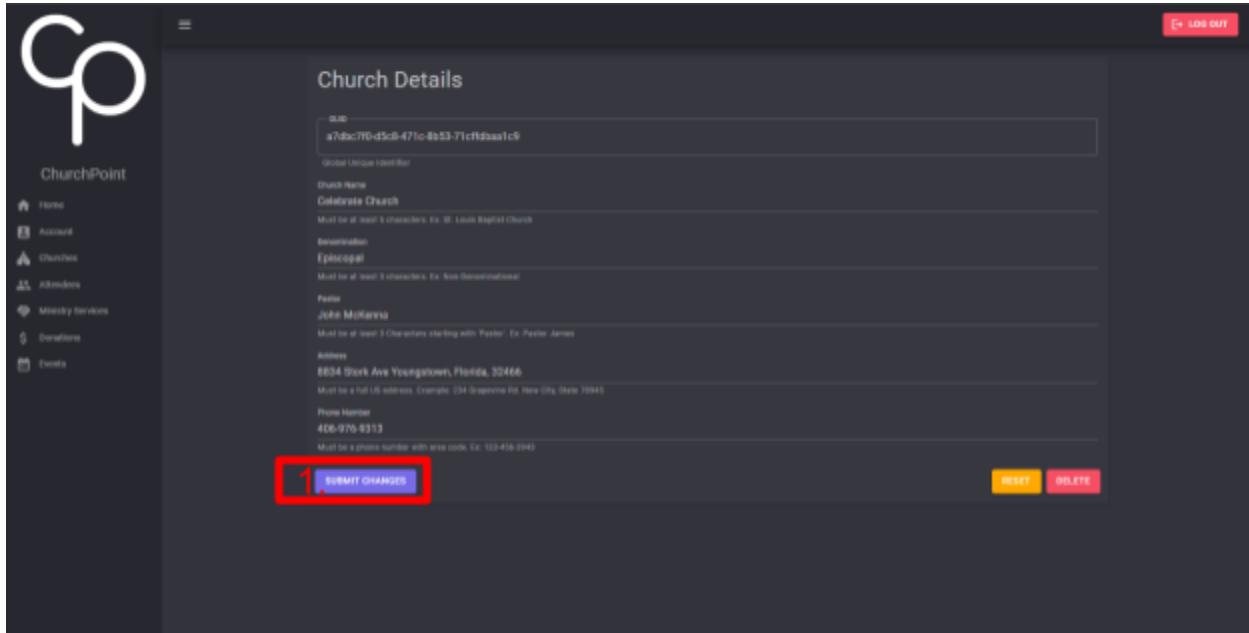
Church ID	Church Name	Denomination	Address	Pastor	Phone Number
12345678-9012-4321-A4B9-B0C0D0E0F0	Bless Hill Refuge	African Methodist Ep.	38 Dingwell St Lowell, Ma	Alexander Brown	215-234-7228
890A9B0C-1234-4567-890A-9876543210	Discipled Church	Orthodox	12380 Fin 1729 Rd Cleveland, Texas, 77328	Ryker Gonzales	262-298-2199

At the bottom right of the table, there is a 'Rows per page' dropdown set to 10, and a navigation bar showing '1-2 of 2' with arrows for navigating through the pages.

Description	To cancel a quick edit refer to 2.3.10, there will be an X icon to the right of the church, click on the icon and the quick edit will be canceled.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i> Then, check the following files: <ol style="list-style-type: none">1. <i>Church.razor</i>2. <i>ChurchBase.cs</i>3. <i>ChurchDetail.razor</i>4. <i>ChurchDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : <ol style="list-style-type: none">1. 10.5 Churches API Functions



9.3.12 Editing Using the Church Detail Page



The screenshot shows the 'Church Details' page in the ChurchPoint software. The page has a dark header with the ChurchPoint logo and a 'LOG OUT' button. On the left is a sidebar with links: Home, Account, Churches, Attendees, Ministry Services, Donations, and Events. The main content area is titled 'Church Details'. It contains several input fields with validation messages: a GUID field ('a7dac7f0-d5cb-471c-8b53-71cf5daa1c9'), a 'Church Name' field ('Celebrate Church'), a 'Denomination' field ('Episcopal'), a 'Pastor' field ('John McCorma'), an 'Address' field ('8834 Stork Ave Youngstown, Florida, 32466'), and a 'Phone Number' field ('406-976-9313'). At the bottom left is a blue 'SUBMIT CHANGES' button, which is highlighted with a red box. At the bottom right are 'RESET' and 'DELETE' buttons.

Description	To edit using the church detail page, refer to 2.3.3 to access the church detail page, and make the edits necessary to the church. Once the edits are sufficient there is a submit changes button on the lower left of the page. Click on it and the changes will be saved.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Church</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Church</i> Then, check the following files: <ol style="list-style-type: none">1. <i>Church.razor</i>2. <i>ChurchBase.cs</i>3. <i>ChurchDetail.razor</i>4. <i>ChurchDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : <ol style="list-style-type: none">1. 10.5 Churches API Functions



9.4 Donations Features

9.4.1 Viewing Donations within the Database

The screenshot shows the ChurchPoint software interface. On the left is a dark sidebar with the ChurchPoint logo and navigation links: Home, Account, Churches (highlighted with a red box), Activities, Ministry Services, Donations, and Events. The main content area has a header 'Churches' with a search bar. Below is a table with columns: Church ID, Church Name, Denomination, Address, Pastor, and Phone Number. A single row is visible: `ecfc1974-fbae-4748-b0f1-60d603ae3200`, **Alpha Church**, Association of Free Lutheran Congregations, 8446 Queens Ct Gates, North Carolina, Alan White, 134-331-6866. At the top right of the table are 'ADD' and 'CREATE' buttons, with 'CREATE' highlighted with a red box and labeled '2.'.

The screenshot shows the 'Select Churches' page. The sidebar includes the ChurchPoint logo and links: Home, Account, Churches (highlighted with a red box), Activities, Ministry Services, Donations, and Events. The main area has a header 'Select Churches' with a search bar. Below is a table with columns: Church ID, Church Name, and Pastor. Five rows are listed: `ecfc1974-fbae-4748-b0f1-60d603ae3200`, **Alpha Church**, Alan White; `1736a6ff-7330-4a6a-8083-c50176e3137b`, **Bless Hill Refuge Church**, Alexander Brown; `162a4e03-ed85-40ce-b0e5-103269eb1d1c`, **Celestology**, Jeremy Lin; `a7fbef70-49a8-475c-8820-71cf8ba1a9`, **Celebrate Church**, John Mekhora; and `68ca3bc1-a1ea-4180-b6ca-fa10cf7f9310`, **Discovered Church**, Rylen Gossles. At the bottom left is a 'Select All' button, highlighted with a red box and labeled '3.'. At the top right is a 'CONFIRM' button, highlighted with a red box and labeled '4.'

The screenshot shows the ChurchPoint software interface with the 'Donations' tab selected. The main area displays a grid of donation records. The columns include: Donation ID, Church Name, Title, Date, Amount, Type, and Actions. The data shows five entries:

Donation ID	Church Name	Title	Date	Amount	Type	Actions
0bca3e16-ef57-4fb3-89d3-8fcd7ef1cfd	Celebration	Donation	5/3/2022 7:24:11 PM -07:00	15	Cash	
0bca3e16-ef57-4fb3-89d3-8fcd7ef1cfd4	Celebration	Donation for Food	5/3/2022 7:24:11 PM -07:00	50	Cash	
6336f749-032a-48be-b5c7-682317957fa	Embrace Gospel Church	Donation	5/3/2022 7:24:11 PM -07:00	100	Cash	
c09e1ac7-93c2-46a7-82db-e60bb37f9679	Celebrate Church	Donation	5/3/2022 7:24:11 PM -07:00	50	Cash	
7ba6f119-9c51-43c2-9bd1-e1edf1b6cf4	Third Experience of Christ Church of the Tree	Services	5/3/2022 7:24:11 PM -07:00	1000	Check	
a1912c49-d210-4309-95a7-1ae94fb0a2b	the River	Payment for Service	5/3/2022 7:24:11 PM -07:00	-43	Check	

Below the grid, there are buttons for 'Rows per page' (10, 20, 50), '14 of 6', and navigation arrows.

Description	<p>To view all Donations in the database using the frontend, follow these directions starting from 1.</p> <ol style="list-style-type: none"> 1. Navigate to Churches. 2. Click Add Churches on the top Right 3. Check “Select All” checkbox 4. Click Confirm on the top right 5. Navigate to the Donations tab and see all of the attendees in the database
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Donation</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Donation</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Donation.razor</i> 2. <i>Donation.Base.cs</i> 3. <i>DonationDetail.razor</i> 4. <i>DonationDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.6 Donation API Functions



9.4.2 Viewing Donations made by Application User

The screenshot shows the ChurchPoint software interface. On the left, there's a sidebar with the ChurchPoint logo and navigation links: Home, Account, Churches, Allocations, and a notifications section showing '1 Donations'. The main area is titled 'Donations' and contains a table with the following data:

Donation ID	Church Name	Title	Date	Amount	Type	Action
5baed1ea-63de-4ca9-7e99-ken08e93	Catawba	Donation	5/3/2022 7:24:11 PM -07:00	10	Cash	⋮
bfc03e15-e657-46db-89d3-ff9d7e81cd4	Catawba	Donation for Fund	5/3/2022 7:24:11 PM -07:00	50	Cash	⋮
6236f745-b22a-48fe-95c7-68231795f1a	Embrace Gospel Church	Donation	5/3/2022 7:24:11 PM -07:00	100	Zelle	⋮
c09efac7-93e2-46a7-62db-ed088373629	Celebrate Church	Donation	5/3/2022 7:24:11 PM -07:00	50	Cash	⋮
fba9116-9e52-45c2-95d3-e1ef193eaf9	Third Experience of Eternity Church of the Triune	Services	5/3/2022 7:24:11 PM -07:00	1000	Check	⋮
a1912c40-c210-4309-95ac-1ee94f50a0b	The River	Payment for Service	5/3/2022 7:24:11 PM -07:00	-40	Check	⋮

At the bottom, it says 'Items per page: 10 < > 1-6 of 6'.

Description	<p>To view all Donations in the Churches the user is a part of, follow these directions starting from 1.</p> <ol style="list-style-type: none"> 1. Navigate to the Donations tab and view all the attendees within the same church
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Donation</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Donation</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Donation.razor</i> 2. <i>Donation.Base.cs</i> 3. <i>DonationDetail.razor</i> 4. <i>DonationDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.6 Donation API Functions



9.4.3 Paging: Setting amount of previous donations to view within Donations List Menu

Donation ID	Church Name	Title	Date	Amount	Type	Action
5baed6d8ac10da4088-7e99feef8e93	Celebration	Donation	5/3/2022 7:24:11 PM -07:00	10	Cash	
b6cde15ed5740db89536fa030f1cd4	Celebration	Donation for food	5/3/2022 7:24:11 PM -07:00	50	Cash	
6336f749a62a480e95c768231795f1a	Embrace Gospel Church	Donation	5/3/2022 7:24:11 PM -07:00	100	Zelle	
c09e1a37-90e2-46a7-82cb-6d889379629	Celebrate Church	Donation	5/3/2022 7:24:11 PM -07:00	50	Cash	
fb94115-9e53-43c2-95d3-e0ef2f8eef11	Third Experience of Diversity Church of the Free	Services	5/3/2022 7:24:11 PM -07:00	1000	Check	
a1912ca9-c010-4309-95a1-1ae94fb6a0b	The River	Payment for Service	5/3/2022 7:24:11 PM -07:00	-40	Check	

Rows per page: 10 | 14 of 6 | < > |

Description	<p>To modify how many donations per page you can view at a time, follow these directions starting from 1.</p> <ol style="list-style-type: none"> In the donations tab, navigate to the bottom right part of the screen and select a value for “Rows per page”
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Donation</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Donation</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> <i>Donation.razor</i> <i>Donation.Base.cs</i> <i>DonationDetail.razor</i> <i>DonationDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 10.6 Donation API Functions



9.4.4 Paging: Changing Between pages within the Donations List Menu

The screenshot shows the ChurchPoint Blazor application's 'Donations' list page. At the top right is a 'LOG OUT' button. Below it is a search bar with a magnifying glass icon and the word 'Search'. To the right of the search bar is a green 'CREATE' button. The main area contains a table with columns: 'Donation ID', 'Church Name', 'Title', 'Date', 'Amount', 'Type', and 'Actions'. Six rows of donation data are listed. At the bottom of the table is a 'Rows per page' dropdown set to '10' and a navigation bar with arrows for '1-6'.

Donation ID	Church Name	Title	Date	Amount	Type	Action
5baed1ea-63de-4ca9-7e99-ken08e93	Cateiology	Donation	5/3/2022 7:24:11 PM -07:00	10	Cash	
bfc03e15-e657-46db-89d3-8fa07e81cd64	Cateiology	Donation for Fund	5/3/2022 7:24:11 PM -07:00	50	Cash	
6236f74b-022a-48fe-95c7-68231795f1fa	Embrace Gospel Church	Donation	5/3/2022 7:24:11 PM -07:00	100	Zelle	
c99f4a74-93c2-46a7-62db-ed088d373629	Celebrate Church	Donation	5/3/2022 7:24:11 PM -07:00	50	Cash	
f6a9116-9e52-45c2-95d3-e1ef193eaf91	Third Experience of Eternity Church of the Brethren	Services	5/3/2022 7:24:11 PM -07:00	1000	Check	
a1912c40-c010-4309-95ac-1ee94ff6a0cb	The River	Payment for Service	5/3/2022 7:24:11 PM -07:00	-40	Check	

Rows per page: 10 < 1-6 > |

Description	<p>To navigate to the next or previous page, follow these steps</p> <ol style="list-style-type: none"> On the bottom right corner of the Events tab, you can see arrows <ol style="list-style-type: none"> Left Bar Arrow - First-page Left Arrow - Previous page Right Arrow - Next page Right Bar Arrow - Last page
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Donation</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Donation</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> <i>Donation.razor</i> <i>Donation.Base.cs</i> <i>DonationDetail.razor</i> <i>DonationDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 10.6 Donation API Functions



9.5 Attendees Features

9.5.1 Viewing Attendees within the Database

The screenshot shows the ChurchPoint software interface. On the left is a sidebar with a logo and navigation links: Home, Account, Churches (highlighted with a red box), Attendees, Ministry services, Donations, and Events. The main content area has a title 'Churches' and a search bar. A table lists one church entry:

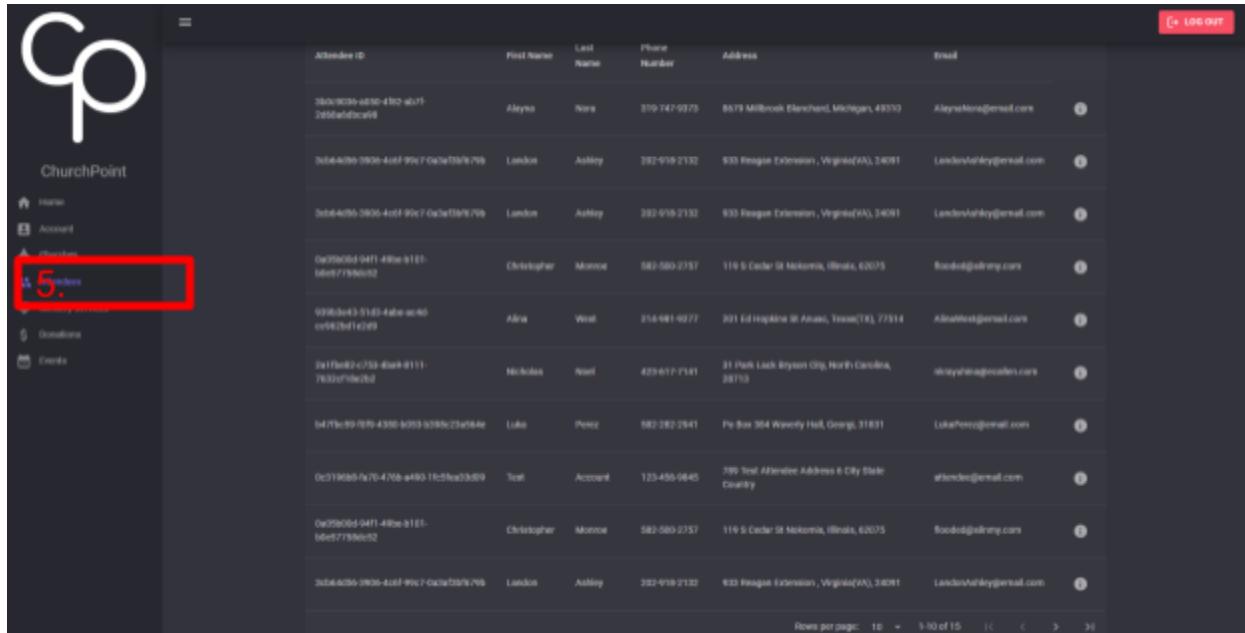
Church ID	Church Name	Denomination	Address	Pastor	Phone Number
estc1974-fabe-4745-bd1f-43d603ec336	Area Church	Association of Free Lutheran Congregations	8446 Quince El Gates, North Carolina, 28645	Alan White	334-331-6856

At the top right are 'LOG OUT' and 'ADD' (purple) and 'CREATE' (green) buttons, both highlighted with a red box. Below the table are pagination controls: 'Rows per page: 10 < > 1-1 of 1'.

The screenshot shows a 'Select Churches' dialog box. On the left is a sidebar with the same navigation links as the previous screen. The main content area has a title 'Select Churches' and a search bar. A table lists five churches:

Church ID	Church Name	Pastor
estc1974-fabe-4745-bd1f-43d603ec336	Area Church	Alan White
1236a6ff-7339-4a8a-8083-03917e31378	Bless Hill Refuge Church	Alexander Brown
162fe632-ed85-40ee-b0a5-103209eb1d1c	Cateology	Jeremy Lin
a78c27b-65a8-4750-6b53-71cf8ea1a9	Celebrate Church	John Mulcahy
8fbca5c1-a7ea-4180-b6cfa-fa10cf1ff910	Discovered Church	Ryker Genesius

At the bottom left is a 'Select All' button highlighted with a red box. At the top right are 'LOG OUT' and 'CONFIRM' (green) buttons, both highlighted with a red box. Below the table are pagination controls: 'Rows per page: 5 < > 1-5 of 10'.



Attendee ID	First Name	Last Name	Phone Number	Address	Email
38d9c9034-ad30-4f82-a671-2d66af3dc494	Aleyya	Nora	319-747-9373	8579 Millbrook Branchard, Michigan, 49310	AleyyaNora@email.com
3c044609-1900-4c6f-99c7-0a3ef2b9f798	Landon	Ashley	302-918-2130	933 Peacock Extension, Virginia(VA), 24091	LandonAshley@email.com
3cb44c55-5b06-4cd1-99c7-0a3ef2b9f798	Landon	Ashley	302-918-2130	933 Peacock Extension, Virginia(VA), 24091	LandonAshley@email.com
0e2fb0d1-04f1-4f0e-b161-bde577956e52	Christopher	Monroe	582-580-2757	119 S Cedar St Melonoma, Illinois, 62075	Foodie@ehome.com
09303e43-5103-4abe-ac40-c9962bd1fc49	Alexa	West	314-681-9277	201 Ed Hopkins St Austin, Texas(TX), 77114	AlexaWest@email.com
2a11fe02-c750-46a9-8111-7633cf0e2d0	Nicholas	Noel	423-617-7141	31 Park Loop Bryson City, North Carolina, 28713	skayakmagazine.com
b479c8970f9-4350-b053-b595c23a564c	Luka	Perez	582-282-2941	P.O. Box 304 Waverly Hall, Georgia, 31881	LukaPerez@email.com
0c019605-fa70-470b-a493-1fcfba39d89	Taylor	Account	123-455-9845	799 Test Attendee Address # City State County	attendee@email.com
0a75b0d1-04f1-4f0e-b161-bde577956e52	Christopher	Monroe	582-580-2757	119 S Cedar St Melonoma, Illinois, 62075	Foodie@ehome.com
3c244609-1900-4c6f-99c7-0a3ef2b9f798	Landon	Ashley	302-918-2130	933 Peacock Extension, Virginia(VA), 24091	LandonAshley@email.com

Description	To view all Attendees in the database using the front-end, follow these directions starting from 1. <ol style="list-style-type: none"> 1. Navigate to Churches. 2. Click Add Churches on the top Right 3. Check “Select All” checkbox 4. Click Confirm on the top right 5. Navigate to the Attendees tab and see all of the attendees in the database
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Attendee</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Attendee</i> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Attendee.razor</i> 2. <i>Attendee.Base.cs</i> 3. <i>AttendeeDetail.razor</i> 4. <i>AttendeeDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.4 Attendees API Functions



9.5.2 Viewing Attendees for Churches assigned to Application User

Attendee ID	First Name	Last Name	Phone Number	Address	Email
78d18039-4f30-43c2-a527-2dd5a6bc5e8	Alyssa	Nora	319-147-9373	8579 Millbrook Blanchard, Michigan, 49310	AlyssaNora@email.com
3d646059-2930-4cf7-9a7c539f979	Landon	Ashley	282-918-2132	933 Pease Extension, Virginia(VA), 24081	LandonAshley@email.com
3cb44659-2930-4cf7-9a7c539f979	Landon	Ashley	282-918-2132	933 Pease Extension, Virginia(VA), 24081	LandonAshley@email.com
0407905d-04f1-490e-91e1-b6e77596e02	Christopher	Monroe	582-580-2757	119 S Cedar St Nokomis, Illinois, 62075	Foodie@cherry.com
99203a43-5103-4ab4-ac46-c5945bf1c69	Alex	West	314-681-9277	301 Ed Hopkins St Austin, Texas(TX), 77114	AlexWest@email.com
2af15e82-c753-4ba9-9111-7633ef1ac20	Nicholas	Noel	425-617-7141	31 Park Luck Bryson City, North Carolina, 28713	NoelNoel@email.com
b479c99-0f99-4390-b053-6395c23e544	Luke	Perez	582-582-2941	Po Box 364 Wreny Hall, Georgia, 31831	LukePerez@email.com
0c31966d-fa70-475b-a493-1fc5fc53e69	Troy	Account	123-456-9845	789 Test Attendee Address 6 City State County	attendee@email.com
0e25b05d-04f1-490e-91e1-b6e77596e02	Christopher	Monroe	582-580-2757	119 S Cedar St Nokomis, Illinois, 62075	Foodie@cherry.com
3d646059-2930-4cf7-9a7c539f979	Landon	Ashley	282-918-2132	933 Pease Extension, Virginia(VA), 24081	LandonAshley@email.com

Rows per page: 10 | 1-10 of 15 | < > >>

Description	To view all Attendees in the Churches the user is a part of, follow these directions starting from 1. <ol style="list-style-type: none">1. Navigate to the Attendees tab and view all the attendees within the same church
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Attendee</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Attendee</i> Then, check the following files: <ol style="list-style-type: none">1. <i>Attendee.razor</i>2. <i>Attendee.Base.cs</i>3. <i>AttendeeDetail.razor</i>4. <i>AttendeeDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : <ol style="list-style-type: none">1. 10.4 Attendees API Functions



9.5.3 Paging: Setting amount of Attendees to view within the Attendees List Menu

The screenshot shows a list of attendees in ChurchPoint. The columns include Attendee ID, First Name, Last Name, Phone Number, Address, and Email. There are 10 rows of data. At the bottom right, there is a pagination control with the number '1' highlighted in a red box, indicating the current page. The control also includes 'Rows per page' and a dropdown menu set to '10'.

Attendee ID	First Name	Last Name	Phone Number	Address	Email
2ba9e036-e018-4f82-8d71-2bb5af5bc043	Aleyne	Nore	319-747-9373	8629 Millbrook (Bentwood), Michigan, 49313	AleyneNore@email.com
2b144636-3906-4e17-99c7-8a3af3b6f79b	Landon	Ashley	202-618-2132	933 Reagan Extension , Virginia(VA), 24091	LandonAshley@email.com
2cb64e36-3906-4ed1-99c7-8a3af3b6f79b	Landon	Ashley	202-618-2132	933 Reagan Extension , Virginia(VA), 24091	LandonAshley@email.com
0a7960d411-49be-b131-b6e77396e02	Christopher	Monroe	582-580-2757	119 S Cedar St Holcomb, Illinois, 62075	freedcd@email.com
09fb3ae3-51d3-4abe-acd4-c9f62bd1c3f9	Alina	West	214-481-4277	201 Ed Hopkins St Austin, Texas(TX), 78714	AlinaWest@email.com
2a1fb3e2-c75d-4ba9-8111-7630cf10e2d2	Nicholas	Neil	423-617-7141	31 Park Lack Bryson City, North Carolina, 28713	nkayahillage@attalla.com
b479e99-f8f9-4383-b083-b398c23e964c	Luka	Perez	582-382-2941	Po Box 384 Waverly Hall, Georgia, 31691	LukaPerez@email.com
0c57960d-fa78-47b9-a940-1fc9fe3d8f09	Trent	Account	123-456-6845	789 Test Attendee Address 6 City State Country	attendee@email.com
0a7960d411-49be-b131-b6e77396e02	Christopher	Monroe	582-580-2757	119 S Cedar St Holcomb, Illinois, 62075	freedcd@email.com
2b144636-3906-4e17-99c7-8a3af3b6f79b	Landon	Ashley	202-618-2132	933 Reagan Extension , Virginia(VA), 24091	LandonAshley@email.com

Description	<p>To modify how many attendees per page you can view at a time, follow these directions starting from 1.</p> <ol style="list-style-type: none"> In the Attendees tab, navigate to the bottom right part of the screen and select a value for "Rows per page"
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Attendee</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Attendee</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> <i>Attendee.razor</i> <i>Attendee.Base.cs</i> <i>AttendeeDetail.razor</i> <i>AttendeeDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 10.4 Attendees API Functions



9.5.4 Paging: Changing Between pages within the Attendees List Menu

The screenshot shows a list of attendees in ChurchPoint. The table columns include Attendee ID, First Name, Last Name, Phone Number, Address, and Email. There are 10 rows of data. At the bottom right of the table, there is a navigation bar with the number '1' in a red box, indicating the current page. Below this, it says 'Rows per page: 10' and shows '1 of 18' with arrows for navigating between pages.

Attendee ID	First Name	Last Name	Phone Number	Address	Email
3ba01603-4310-4f62-8f27-2d5f6af5ca98	Aleyne	Nora	319-147-9373	8679 Millbrook Blanchard, Michigan, 48313	AleyneNora@email.com
3b644676-3906-4c61-9917-6a3af3b46790	Landon	Ashley	202-918-2132	933 Reagan Extension, Virginia(VA), 24061	LandonAshley@email.com
3eb44676-3906-4c61-99c7-6a3af3b4679b	Landon	Ashley	202-918-2132	933 Reagan Extension, Virginia(VA), 24061	LandonAshley@email.com
0a025908-04f1-49be-b101-b4e577596f52	Christopher	Monroe	582-580-2757	119 S Cedar St Holomis, Illinois, 62075	Booboo@lenny.com
9393e343-5103-4ab6-a6d4-cc962bd1fd97	Alysa	West	214-681-4277	201 Ed Hopkins St Austin, Texas(TX), 78714	AlysaWest@email.com
3a115e02-c75d-4b9b-b111-7630cf10e2b2	Nicholas	Noah	429-617-7141	31 Park Lack Bryson City, North Carolina, 28713	noahyakimagination.com
b479bc99-7f89-4368-bc83-b398c23e504c	Luka	Perez	582-282-2941	Po Box 384 Mewery Hall, Georgia, 31631	LukaPerez@email.com
0c3196a6-fa78-476b-a90-1fc5fe336809	Tier	Acosta	123-456-9845	769 Test Attendee Address 6 City State Country	attendee@email.com
0a025908-04f1-49be-b101-b4e577596f52	Christopher	Monroe	582-580-2757	119 S Cedar St Holomis, Illinois, 62075	Booboo@lenny.com
3b246036-3906-4c61-99c7-6a3af3b4679b	Landon	Ashley	202-918-2132	933 Reagan Extension, Virginia(VA), 24061	LandonAshley@email.com

Description	To view more pages of the list of attendees, follow these directions starting from 1. 1. On the bottom right corner of the Attendees tab, click the right arrow to go to the next page. a. The right arrow with the bar will send you to the last page.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Attendee</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Attendee</i> Then, check the following files: 1. <i>Attendee.razor</i> 2. <i>Attendee.Base.cs</i> 3. <i>AttendeeDetail.razor</i> 4. <i>AttendeeDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : 1. 10.4 Attendees API Functions



9.6 Events Features

9.6.1 Viewing Events within the Database

The screenshot shows the ChurchPoint software interface. On the left is a sidebar with a logo and navigation links: Home, Account, Churches (highlighted with a red box), Attendees, Ministry Services, Donations, and Events. The main content area has a title 'Churches' and a search bar. A table lists one church entry:

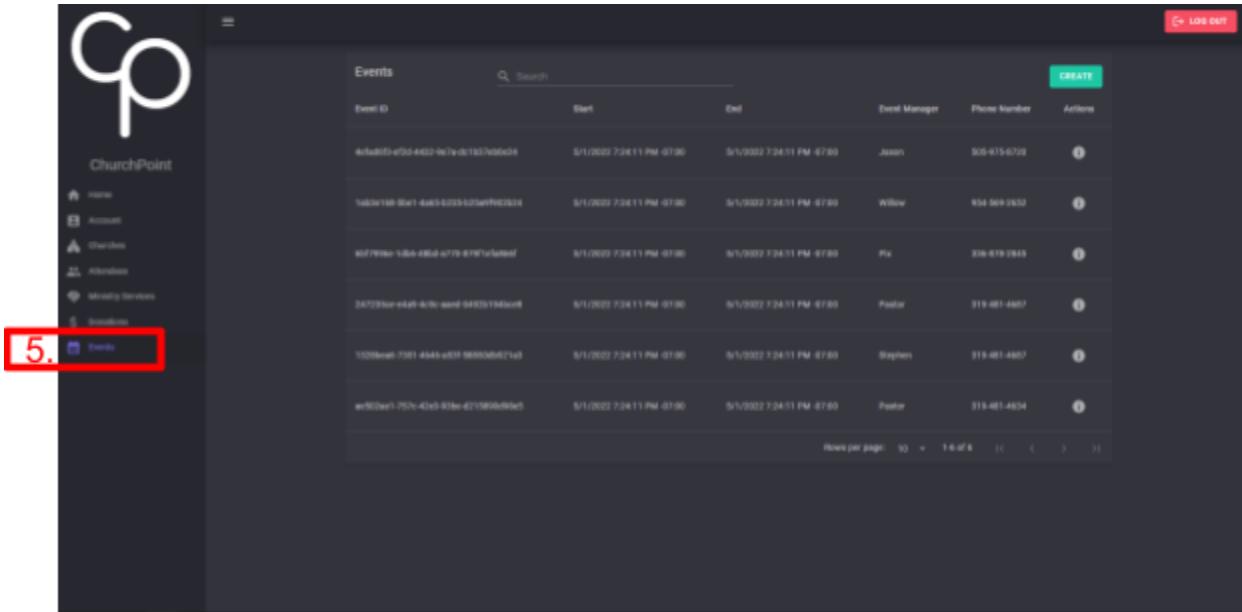
Church ID	Church Name	Denomination	Address	Pastor	Phone Number
ecfc1974-fabe-4748-b0f1-63d603ec3200	Alpha Church	Association of Free Lutheran Congregations	8446 Queens Ct Gates, North Carolina, 28640	Alan White	134-331-6666

At the top right are 'ADD' and 'CREATE' buttons, with '2.' numbered above them. At the bottom right are pagination controls.

The screenshot shows the 'Select Churches' page. The sidebar is identical to the previous screen. The main content area has a title 'Select Churches' and a search bar. A table lists five churches, with the last one highlighted by a red box and labeled '3.'. Below the table is a 'Select All' checkbox labeled '4.' with a red box around it.

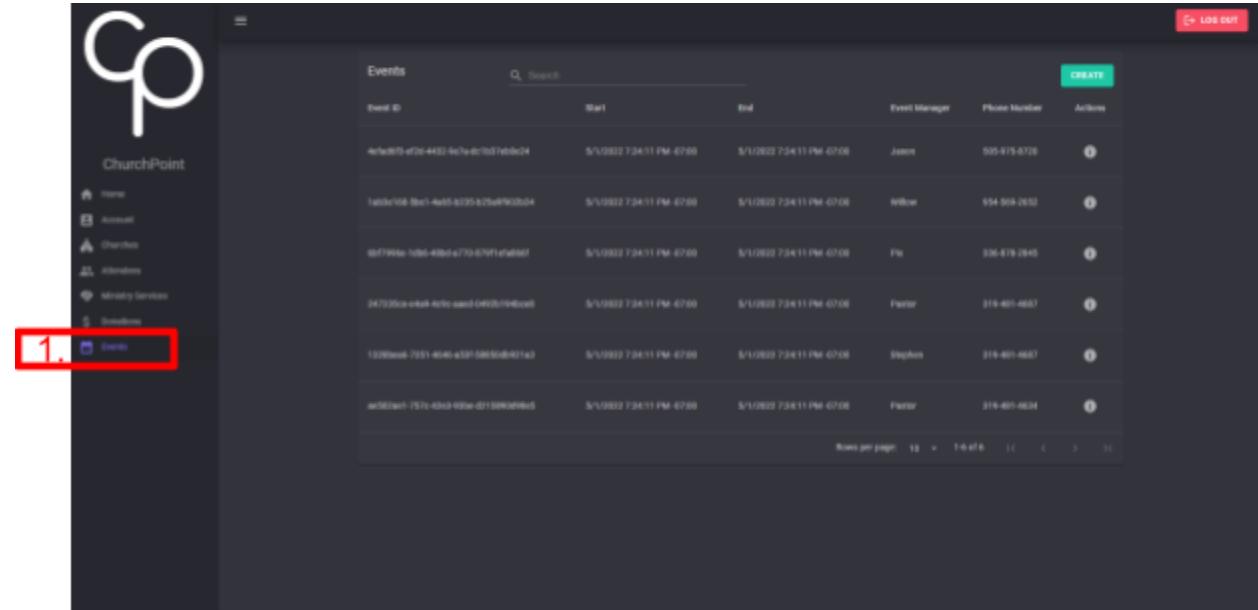
Church ID	Church Name	Pastor
ecfc1974-fabe-4748-b0f1-63d603ec3200	Alpha Church	Alan White
12345678-1234-4567-8900-123456789012	Bless Hill Refuge Church	Alexander Brown
12345678-9876-5432-1234-123456789012	Celestology	Jeremy Lin
45678901-2345-6789-1234-567890123456	Celebrate Church	John Mulcahy
56789012-3456-7890-1234-567890123456	Discovered Church	Ryder Gennales

At the top right are 'CONFIRM' and 'CANCEL' buttons, with '4.' numbered above them. At the bottom right are pagination controls.



Description	<p>To view all Events in the database using the frontend, follow these directions starting from 1.</p> <ol style="list-style-type: none"> 1. Navigate to Churches tab 2. Click Add on the top right corner 3. Check the Select All checkbox 4. Click Confirm 5. Click Events Tab to view all events
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Events</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Events</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Event.razor</i> 2. <i>EventBase.cs</i> 3. <i>EventDetail.razor</i> 4. <i>EventDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.7 Event Features

9.6.2 Viewing Events for Churches assigned to Application User



The screenshot shows the ChurchPoint software interface. On the left, there is a dark sidebar with the ChurchPoint logo at the top. Below the logo are several menu items: Home, Account, Churches, Addresses, Ministry Services, and Schedules. The 'Events' item is highlighted with a red box and a number '1.' next to it. To the right of the sidebar is a large table titled 'Events'. The table has columns for Event ID, Start, End, Event Manager, Phone Number, and Actions. There are six rows of event data listed. At the bottom of the table, there is a pagination control labeled 'Events per page' with a dropdown set to '10', and a page indicator '1-6 of 6'.

Event ID	Start	End	Event Manager	Phone Number	Actions
4fbcd0f3-e0e4-4432-8e7e-dc1037eb8c24	5/5/2022 7:34:11 PM -07:00	5/1/2022 7:34:11 PM -07:00	Jason	505-875-0726	⋮
14033e168-5b01-4e05-8395-625e8f9c0204	5/5/2022 7:34:11 PM -07:00	5/1/2022 7:34:11 PM -07:00	Willow	654-568-2652	⋮
6d799de1-1dbd-48bd-a710-87bfefabedf	5/5/2022 7:34:11 PM -07:00	5/1/2022 7:34:11 PM -07:00	Pete	319-419-2845	⋮
347339cc-e4a9-4f90-aed0-09058794bc00	5/5/2022 7:34:11 PM -07:00	5/1/2022 7:34:11 PM -07:00	Peter	319-401-4607	⋮
13200ea6-7951-4046-a501-0865ab901a3	5/5/2022 7:34:11 PM -07:00	5/1/2022 7:34:11 PM -07:00	Stephen	319-401-4607	⋮
aef039e1-757c-42a9-900a-021180c5b4a5	5/5/2022 7:34:11 PM -07:00	5/1/2022 7:34:11 PM -07:00	Peter	319-401-4604	⋮

Description	<p>To view all Events from the Church(es) assigned to the user using the frontend, follow these directions starting from 1.</p> <ol style="list-style-type: none"> Click Churches to view all available events
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Events</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Events</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> <i>Event.razor</i> <i>EventBase.cs</i> <i>EventDetail.razor</i> <i>EventDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 10.7 Event Features

9.6.3 Paging : Setting amount of Events to view within the Events List Menu

Event ID	Start	End	Event Manager	Phone Number	Actions
4ef4d4f3-e72e-4432-9c7e-0183764de24	5/1/2022 7:24:11 PM -07:00	5/1/2022 7:24:11 PM -07:00	Jesse	809-979-8739	Edit
1ab53e168-5be1-4a65-b235-b25e9903004	5/1/2022 7:24:11 PM -07:00	5/1/2022 7:24:11 PM -07:00	Wilson	854-554-2852	Edit
587f946e-1d86-48d4-4729-87ff1fa8d6f	5/1/2022 7:24:11 PM -07:00	5/1/2022 7:24:11 PM -07:00	Pix	336-870-2845	Edit
247238ee-64b9-4c80-aed3-048251946e8	5/1/2022 7:24:11 PM -07:00	5/1/2022 7:24:11 PM -07:00	Pastor	319-481-4867	Edit
1320fbca-7311-4448-a53f-58810a8921ad	5/1/2022 7:24:11 PM -07:00	5/1/2022 7:24:11 PM -07:00	Sophie	319-481-4867	Edit
ae502ee1-787c-43c3-98be-d21580a696e5	5/1/2022 7:24:11 PM -07:00	5/1/2022 7:24:11 PM -07:00	Pastor	319-481-4864	Edit

Rows per page: 10 ▾ 6 of 6 | < < > >

Description	<p>To set a limit on how many events to view per page, follow these steps.</p> <ol style="list-style-type: none"> On the bottom right corner of the page, click the Rows per page dropdown and select a value
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Events</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Events</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> <i>Event.razor</i> <i>EventBase.cs</i> <i>EventDetail.razor</i> <i>EventDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 10.7 Event Features



9.6.4 Paging : Changing Between pages within the Events List Menu

The screenshot shows the ChurchPoint software interface. On the left is a sidebar with the ChurchPoint logo and links to Home, Account, Churches, Attendees, Ministry Services, Donations, and Events. The main area is titled "Events" and contains a table with columns: Event ID, Start, End, Event Manager, Phone Number, and Actions. There are seven rows of event data. At the bottom of the table is a navigation bar with the text "Rows per page: 10" and a dropdown menu set to "9 of 1". To the right of this are arrows for navigating between pages: a left arrow, a right arrow, and a double-right arrow.

Description	<p>To navigate to the next or previous page, follow these steps</p> <ol style="list-style-type: none"> 1. On the bottom right corner of the Events tab, you can see arrows <ol style="list-style-type: none"> a. Left Bar Arrow - First page b. Left Arrow - Previous page c. Right Arrow - Next page d. Right Bar Arrow - Last page
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>Events</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>Events</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. <i>Event.razor</i> 2. <i>EventBase.cs</i> 3. <i>EventDetail.razor</i> 4. <i>EventDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.7 Event Features

9.7 Ministry Service Features

9.7.1 Viewing Ministry Service within the Database

The screenshot shows the ChurchPoint software interface. On the left is a sidebar with a logo and navigation links: Home, Account, Churches (which is highlighted with a red box), Attendees, Ministry services, Donations, and Events. The main content area has a title 'Churches' and a search bar. A table lists one church entry:

Church ID	Church Name	Denomination	Address	Pastor	Phone Number
estc1974-fabe-4745-bd8f-63d603ec336	Area Church	Association of Free Lutheran Congregations	8446 Querencia Ct Gates, North Carolina, 27526	Alan White	334-331-6856

At the top right are 'LOG OUT' and 'ADD' (purple) and 'CREATE' (green) buttons, both enclosed in a red box. At the bottom right are pagination controls: 'Rows per page: 10', '1-1 of 1', and arrows.

The screenshot shows the 'Select Churches' page. The sidebar is identical to the previous screen. The main content area has a title 'Select Churches' and a search bar. A table lists multiple churches:

Church ID	Church Name	Pastor
estc1974-fabe-4745-bd8f-63d603ec336	Area Church	Alan White
1236a6ff-7339-4a8a-8083-039176e31376	Bless Hill Refuge Church	Alexander Brown
162fe632-ed85-40ee-b0a5-103209eb1d1c	Cateology	Jeremy Lin
a78c27b-65a8-4750-6b53-71cf8a1a9	Celebrate Church	John Mulcahy
8fbca5c1-a7ea-4180-b6fa-fa10cf1ff910	Discovered Church	Ryker Genesius

At the top right are 'LOG OUT' and 'CONFIRM' (green) buttons, both enclosed in a red box. At the bottom right are pagination controls: 'Rows per page: 10', '1-5 of 10', and arrows. A red box labeled '3. Select All' covers the bottom-left corner of the table.

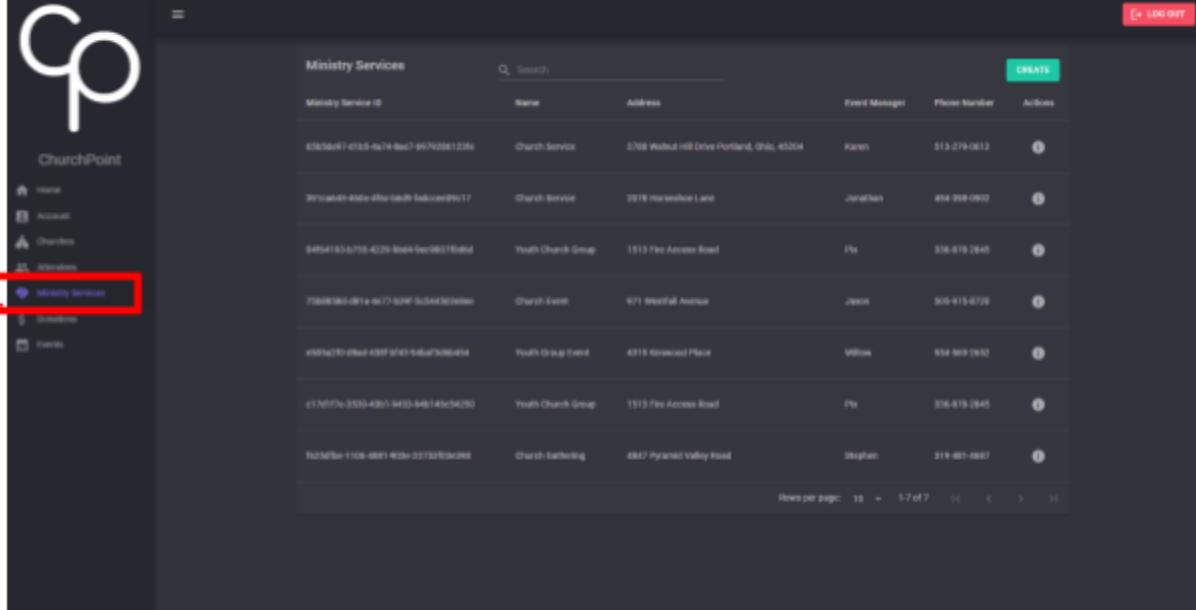
Ministry Services					
Ministry Service ID	Name	Address	Event Manager	Phone Number	Actions
60f8d6e7-d1b5-4a74-8ac7-4979085122fe	Church Service	3788 Walnut Hill Drive Portland, Ohio, 45034	Karen	513-279-0812	
591ca045-46da-4ff0-96a9-fa0cc079c17	Church Service	2018 Horsehair Lane	Jonathan	484-589-0902	
04564193-8755-4229-9bd4-9ec90879b0d	Youth Church Group	1513 Five Acres Road	Pix	330-670-1845	
7f068564-8f1a-4c77-b09f-9044803268ee	Church Event	971 Westfall Avenue	Jason	809-973-6720	
e050a292-86ed-438f-b113-446e35d6454	Youth Group Event	4219 Remond Place	Wilson	854-589-2652	
c1760f7a-9830-4321-8432-94b14b504290	Youth Church Group	1513 Five Acres Road	Pix	330-670-1845	
f025efbe-1186-4881-983e-227593e598	Church Gathering	4047 Pyramid Valley Road	Stephen	319-481-4637	

Rows per page: 10 ▾ 1 2 3 4 5 6 7 > >>

Description	<p>To view all Ministry Services in the database using the frontend, follow these directions starting from 1.</p> <ol style="list-style-type: none"> 1. Navigate to Churches tab 2. Click Add on the top right corner 3. Check the Select All checkbox 4. Click Confirm 5. Click Ministry Services to view all Ministry Services in the database
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>MinsitryService</i></p> <p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>MinsitryService</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 9. <i>MinistryService.razor</i> 10. <i>MinistryServicesBase.cs</i> 11. <i>MinistryServiceDetail.razor</i> 12. <i>MinistryServiceDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.8 Ministry API Functions



9.7.2 Viewing Ministry Service for Churches assigned to Application User



The screenshot shows the ChurchPoint application interface. On the left, there is a dark sidebar with the ChurchPoint logo at the top. Below the logo are several navigation items: Home, Assets, Churches, Ministries, Ministry Services (which is highlighted with a red box), Ministries, Events, and Reports. The main content area is titled "Ministry Services". It features a search bar with a magnifying glass icon and a "CREATE" button. A table lists seven ministry services, each with a "Details" button. The columns are: Ministry Service ID, Name, Address, Event Manager, Phone Number, and Actions. The data in the table includes:

Ministry Service ID	Name	Address	Event Manager	Phone Number	Actions
6103de87-d105-4a74-bac7-497c008120fe	Church Service	3708 Walnut Hill Drive Portland, Ohio, 45204	Karen	513-279-0612	
391ca4d3-40fa-4fbc-ba0e-99c17	Church Service	3018 Mansfield Lane	Jonathan	414-959-0912	
04041b3d-7155-4229-8ab4-bee9602710d4	Youth Church Group	1513 Elm Access Road	Pete	334-879-2845	
750003b3-c81a-4c77-939f-0c54d36c0ee	Church Event	671 Marshall Avenue	Jacob	505-419-8728	
e17b0f7c-0530-43b1-9423-94b14fc54290	Youth Group Event	4819 Kinnelon Place	Willow	914-819-7812	
b2ad5a-1108-4891-402a-2213d8a0e001	Youth Church Group	1513 Elm Access Road	Pete	334-879-2845	
48a0fba-1108-4891-402a-2213d8a0e001	Church Gathering	4847 Pyramid Valley Road	Stephen	219-481-6607	

At the bottom of the table, there are buttons for "Rows per page" (set to 10), "1-7 of 7", and navigation arrows.

Description	To view all Ministry Services from the Church(es) assigned to the user using the frontend, follow these directions starting from 1. 1. Click on the Ministry Services tab to view information
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>MinsitryService</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>MinsitryService</i> Then, check the following files: 1. <i>MinistryService.razor</i> 2. <i>MinistryServicesBase.cs</i> 3. <i>MinistryServiceDetail.razor</i> 4. <i>MinistryServiceDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : 1. 10.8 Ministry API Functions



9.7.3 Paging: Setting amount of Ministry Service to view within the Ministry Service List Menu

The screenshot shows a list of ministry services in ChurchPoint. At the bottom right of the list table, there is a dropdown menu labeled "Rows per page" with the value "10". A red box highlights this dropdown.

Ministry Services						<input type="button" value="CREATE"/>
Ministry Service ID	Name	Address	Event Manager	Phone Number	Actions	
6081de47-d155-4a74-8ac7-4970d85122fe	Church Service	3788 Walnut Hill Drive Portland, Ohio, 45204	Karen	513-279-0812		
591c9d45-46da-4fbc-96e9-fadcc079c17	Church Service	2018 Horseshoe Lane	Jonathan	484-589-0902		
04164193-8755-4229-9b4d-4ee90087b004	Youth Church Group	1513 Five Acceso Road	Pix	330-670-1845		
70068564-0f1a-4c77-b39f-9c54400216ee	Church Event	971 Westfall Avenue	Jason	809-975-6720		
e025a293-9ed4-430f-b143-94baef3d64f4	Youth Group Event	4219 Kemwood Place	Wilow	554-589-2652		
c1760f7a-9830-4301-9432-9481f4850420	Youth Church Group	1513 Five Acceso Road	Pix	330-670-1845		
f025dfb1-1166-4681-903e-227593e098	Church Gathering	4047 Pyramid Valley Road	Stephen	319-481-4687		

Description	To set a limit on how many ministry services to view per page, follow these steps. 1. On the bottom right corner of the page, click the Rows per page dropdown and select a value
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>MinsitryService</i> <i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>MinsitryService</i> Then, check the following files: 13. <i>MinistryService.razor</i> 14. <i>MinistryServicesBase.cs</i> 15. <i>MinistryServiceDetail.razor</i> 16. <i>MinistryServiceDetailBase.cs</i> If other problems persist, check the following functions pertaining to the following sections in the API section : 1. 10.8 Ministry API Functions



9.7.4 Paging: Changing Between pages within the Ministry Service List Menu

Ministry Services						<input type="text" value="Search"/>	<button>CREATE</button>
Ministry Service ID	Name	Address	Event Manager	Phone Number	Actions		
6083de47-d155-4a74-8ac7-497028d122fe	Church Service	3788 Walnut Hill Drive Portland, Ohio, 45204	Karen	513-279-0812			
591caef4-468a-4fbc-96e9-fadcc79c717	Church Service	2018 Marquette Lane	Jonathan	484-595-0902			
04f64193-8755-4229-9b84-49e9087930d	Youth Church Group	1513 Five Acres Road	Pix	330-670-2845			
7006564-8f1a-4c77-b29f-bc54493269ee	Church Event	971 Westfield Avenue	Jason	859-979-8720			
e925a293-d1ed-430f-b143-94baef3d64f4	Youth Group Event	4319 Kemper Place	Wilson	514-569-2652			
c176ff7a-9830-4301-9413-94f146104290	Youth Church Group	1513 Five Acres Road	Pix	330-670-2845			
f225efbe-1166-4881-903e-2275f93e098	Church Gathering	4047 Pyramid Valley Road	Stephen	319-481-4637			

Show per page: 10 < > 1 of 1

Description	<p>To navigate to the next or previous page, follow these steps</p> <ol style="list-style-type: none"> 1. On the bottom right corner of the Events tab, you can see arrows <ol style="list-style-type: none"> a. Left Bar Arrow - First page b. Left Arrow - Previous page c. Right Arrow - Next page d. Right Bar Arrow - Last page
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to <i>ChurchPoint.Blazor.Web</i> → <i>Components</i> → <i>ListPages</i> → <i>MinsitryService</i></p> <p><i>ChurchPoint.Blazor.Web</i> → <i>Component</i> → <i>DetailPages</i> → <i>MinsitryService</i></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 17. <i>MinistryService.razor</i> 18. <i>MinistryServicesBase.cs</i> 19. <i>MinistryServiceDetail.razor</i> 20. <i>MinistryServiceDetailBase.cs</i> <p>If other problems persist, check the following functions pertaining to the following sections in the API section:</p> <ol style="list-style-type: none"> 1. 10.8 Ministry API Functions



10.0 ChurchPoint Web API

10.1 Open API Swagger Documentation Site General Features

10.1.1 Getting swagger.json document

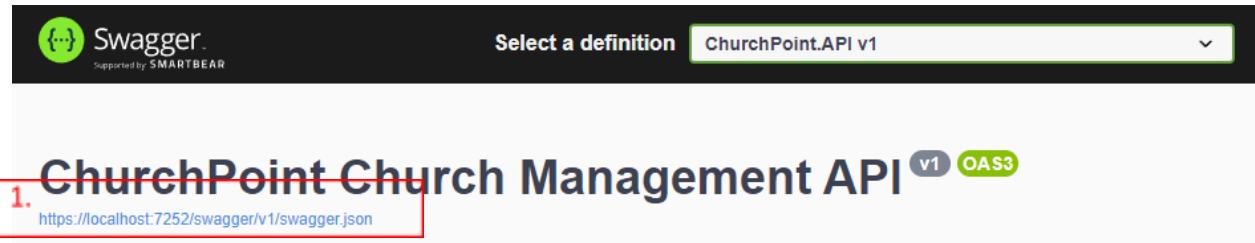


Figure 1 API: Swagger Documentation Title Page

```
{  
  "openapi": "3.0.1",  
  "info": {  
    "title": "ChurchPoint Church Management API",  
    "version": "v1"  
  },  
  "paths": {  
    "/api/AppUser/GetApplicationUserById/{id)": {  
      "get": {  
        "tags": [  
          "AppUser"  
        ],  
        "parameters": [  
          {  
            "name": "id",  
            "in": "path",  
            "required": true,  
            "schema": {  
              "type": "string",  
              "format": "uuid"  
            }  
          }  
        ]  
      }  
    }  
  }  
}
```

Figure 2 API: Output of Swagger JSON API Documentation

Description	The retrieval of the swagger.json file can be attained by clicking on the following link on the swagger website. From there it will take you to the document as seen in <i>figure 2 API</i>
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → <i>ChurchPoint.API</i> Then, check the following files:

	<p>1. Program.cs</p> <p>Make sure the <code>app.UseSwagger()</code>, <code>app.UseSwaggerUI()</code>, and <code>builder.Services.AddSwaggerGen()</code> functions are present within the program.</p>
--	---

10.1.2 Selecting API Definition



Figure 3 API: Swagger Open API Header

Description	On the top right corner of the Swagger Open API User Interface you will see a selection for the AP Definition
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → <code>ChurchPoint.API</code></p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Program.cs <p>Make sure the <code>app.UseSwagger()</code>, <code>app.UseSwaggerUI()</code>, and <code>builder.Services.AddSwaggerGen()</code> functions are present within the program.</p>

10.1.3 Looking at API Schemas



Figure 4 API: Bottom of Swagger API UI

```

2. AddAttendeeToChurchByIdCommand {
    attendeeId string($uuid)
    churchId string($uuid)
}

```

Figure 5 API: Expanded Schema View

Description	To look at the API Schemas once in the Swagger Open API User Interface there Will be a section labeled schemas (<i>figure 4 API</i>). Once clicked a drop-down will appear below with the schemas (<i>figure 5 API</i>).
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → <i>ChurchPoint.API</i> AND src → Core → ChurchPoint.Application & ChurchPoint.Domain</p> <p>Then, check the following files:</p> <ol style="list-style-type: none"> 1. Program.cs 2. In ChurchPoint.Application → Models AND ChurchPoint.Application → Profiles <ul style="list-style-type: none"> a. MappingProfile.cs b. Features (check all Commands and Queries) 3. In ChurchPoint.Domain → Entities <ul style="list-style-type: none"> a. Attendee.cs b. AttendeeType.cs c. Church.cs d. Donation.cs e. Event.cs f. MinistryService.cs

10.2 API Authorize Access w/ Bearer Token

10.2.1 Application User Authentication Register

AppUserAuthentication	
POST	/api/AppUserAuthentication/Authenticate
1. POST	/api/AppUserAuthentication/Register



POST /api/AppUserAuthentication/Register

Parameters

No parameters

Request body

3. Try it out

2. application/json

Request body

application/json

```
{  
    "firstName": "James Berk",  
    "lastName": "Matthews",  
    "email": "jmatthews@churchnewport.org",  
    "userName": "JamesBerkAtNewPort",  
    "password": "JBM@Password1!"  
}
```

4.

5. Execute

The screenshot shows a REST API endpoint for user registration. The endpoint is POST /api/AppUserAuthentication/Register. The interface includes sections for Parameters (none), Request body (application/json), and Request body content (JSON object). Step 3 highlights the 'Try it out' button, step 2 highlights the 'application/json' dropdown, step 4 highlights the JSON request body, and step 5 highlights the 'Execute' button.

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7252/api/AppUserAuthentication/Register' \
  -H 'accept: text/plain' \
  -H 'Content-Type: application/json' \
  -d '{
    "firstName": "James Berk",
    "lastName": "Matthews",
    "email": "jbmattew@churchnewport.org",
    "userName": "JamesBerkAtNewPort",
    "password": "JBM@Password1!"
}'
```

Request URL

<https://localhost:7252/api/AppUserAuthentication/Register>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "userId": "7b80917f-26ef-4b3d-bbb5-8e970faaa34b" }</pre> <p>6.</p> <p>Download</p> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Sat, 30 Apr 2022 22:13:31 GMT server: Kestrel</pre>

Description	<p>Testing a new application user. Refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Go to the AppUserAuthentiaktion Section of the API Website 2. Make sure the parameter drop-down is set to “application/json” 3. Click on “Try it out” 4. Within the text box fill in with the specified test input shown below 5. Click on “Execute” 6. You should get the Expected Output Shown Below with a Code denoting 200 OK
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserAuthenticationController.cs <p>Application Layer</p> <p>src → Core → Contracts → Identity</p> <p>src → Core → Models → Authentication</p>



Then look at the following files:

1. IAuthenticationService.cs
2. RegistrationRequest.cs
3. RegistrationResponse.cs
4. JwtSettings.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence
src → Infrastructure → ChurchPoint.Persistence → Services

Then look at the following files:

1. AuthenticationService.cs
2. PersistenceServiceRegistration.cs

10.2.2 Application User Authentication Authenticate

AppUserAuthentication

1. POST /api/AppUserAuthentication/Authenticate

POST /api/AppUserAuthentication/Register

POST /api/AppUserAuthentication/Authenticate

Parameters

2. Try it out



Parameters

No parameters

Request body application/json

```
{
  "email": "string",
  "password": "string"
}
```

4. Execute

Code **Details**

200

Response body

```
{
  "id": "91d461a3-e115-4b95-91d4-12a888cia4ed",
  "userName": "firstlastName",
  "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJUZXN0VXNlcjEiLCJqdGkiOiI2MmF1ZGU5O51jNzg0LTQ5MzEtODc3ZC0xYTWjZTA1Y2V0Nzk1LCC1bWfpbC16lnVzZJAZXhbXBsZS5jb20iLCJiaMQiOiISMMQ8NjFnMy1MTExTR10TUtOTfKNC0xMmE4ODhjMWE0ZMQLC1eHAjD1E2NTExNTEyMjMwciImlzcyI6IkNodKJjaFBvaW5OSWR1bmRpdiKilCJhdqioiJ0aiIVyY2hQb21udtikZW50aXR5VXN1ciJ9.71vUzhilwHiD1TL-JK5aK2fHfAu0q" Download
}
```

Response headers

```

cache-control: no-cache,no-store
content-type: application/json; charset=utf-8
date: Sat,30 Apr 2022 22:27:22 GMT
expires: Thu,01 Jan 1970 00:00:00 GMT
pragma: no-cache
server: Kestrel
x-firebase-spdy: h2

```

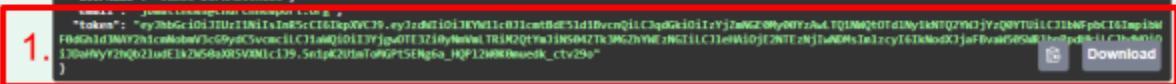
5.

Description	To authenticate the user application refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Go to the AppUserAuthentiaktion Section of the API Website 2. Under post make sure there is a try it out button off the right 3. Once clicked, within the text box go ahead and put the credentials used for the existing user. 4. Execute the command 5. If done correctly a 200 ok response should be shown with the auto-generated token that will be used for authorization.
--------------------	--



List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserAuthenticationController.cs <p>Application Layer</p> <p>src → Core → Contracts → Identity</p> <p>src → Core → Models → Authentication</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAuthenticationService.cs 2. AuthenticationRequest.cs 3. AuthenticationResponse.cs 4. JwtSettings.cs <p>Infrastructure Layer</p> <p>src → Infrastructure → ChurchPoint.Persistence</p> <p>src → Infrastructure → ChurchPoint.Persistence → Services</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AuthenticationService.cs 2. PersistenceServiceRegistration.cs
---------------------------	---

10.2.3 Authorize

Code	Details
200	<p>Response body</p> <pre>{ "id": "7008917f-26ef-4b3d-8b65-8e970faaa34b", "userName": "James BerkMatthews", "tokens": ["eyJhbGciOiJIUzI1NiJzImR5CCEIiip0WCIJ.yJzadHtDjL3KYNdlcB31cmBd51d3BcnqjILC3qdGkjdIzYjZmNGZ0My00YzA4LQ1NMjeOTdINyEhNTQ2YnCjYxQNYT01LC31NFphCT8ImjMFR4Gh1d3MhV211cnMhMh073c69ydC5vncnLLC1iaQj0113YJg-e0TE3z10yMhMhLTr1R2Q1y31NSh4ZTk3M627hYhErNG1LLC1eMh0jE2NTEzNjJwMhMsIm1zcy161h0oX7jaFhaw5h5w811x0pdhLLC1h4n10138wHyr2hQq21udef1h2h5baXRS5VhNLc139_5m1pk2utnTo9GPtSENg5o_HQPl2HhR0newek_k_ctv29a"] }</pre> <p>1.  </p>



Swagger
powered by SMARTBEAR

Select a definition ChurchPoint.API v1

ChurchPoint Church Management API v1 OAS3

<https://localhost:7252/swagger/v1/swagger.json>

2. Authorize 

AppUser

Available authorizations

Bearer (apiKey)

JWT Authorization header using the Bearer scheme. Enter 'Bearer' [space] and then your token in the text input below. Example: Bearer 12345abcdef

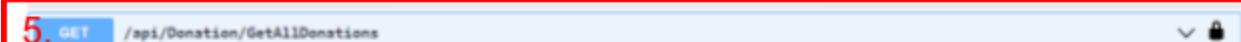
Name: Authorization
In: header
Value:

3. Bearer eyJhbGciOiJIUzI1NiJ9 

4. Authorize 

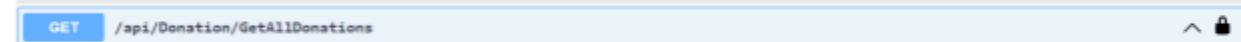
Try it out 

Donation

5. GET /api/Donation/GetAllDonations 

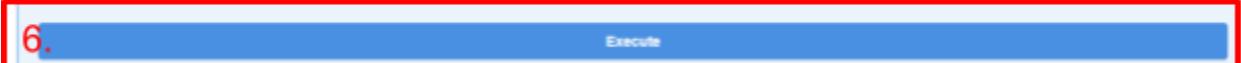
Cancel 

Donation

GET /api/Donation/GetAllDonations 

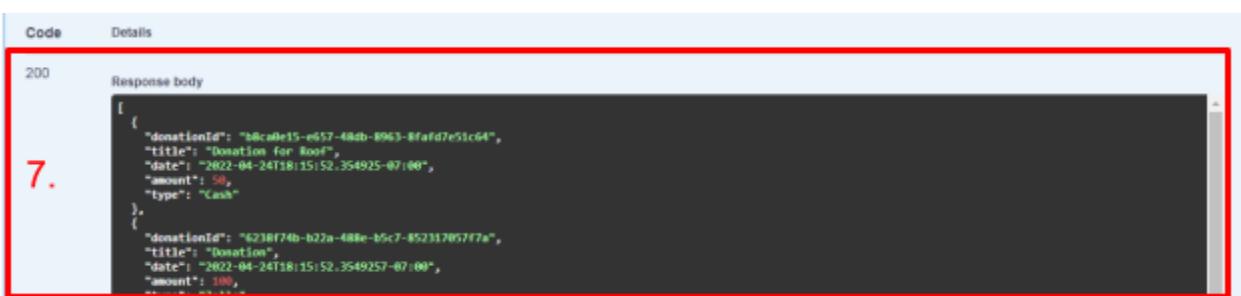
Parameters 

No parameters 

6. Execute 

Code Details

200 Response body

7. 

```
[
  {
    "donationId": "8bcade15-e657-48db-8963-8ffad7e51c64",
    "title": "Donation for Root",
    "date": "2022-04-24T18:15:52.354925-07:00",
    "amount": 50,
    "type": "Cash"
  },
  {
    "donationId": "623bf74b-b22a-488e-b5c7-852317857f7a",
    "title": "Donation",
    "date": "2022-04-24T18:15:52.3549257-07:00",
    "amount": 100,
    "type": "Cash"
  }
]
```



Description	<p>To authorize application user refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Make sure to do 3.2.2 to copy the user token from the 200 OK response 2. Navigate to the top of the API UI and click on Authorize 3. Input the “Bearer” and then paste in the auto generate token from 3.2.2 and hit “Close” 4. Click on “Authorize” 5. Navigate to “Donation” and click to expand drop-down 6. Click “Execute” 7. 200 OK should be returned with list of donations in the Response Body
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserAuthenticationController.cs <p>Application Layer src → Core → Contracts → Identity src → Core → Models → Authentication</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAuthenticationService.cs 2. AuthenticationRequest.cs 3. AuthenticationResponse.cs 4. JwtSettings.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence src → Infrastructure → ChurchPoint.Persistence → Services</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AuthenticationService.cs 2. PersistenceServiceRegistration.cs



10.3 Application User API Functions

10.3.1 Get Application User By Id

The screenshot shows a REST API documentation interface for the 'AppUser' resource. The 'Get ApplicationUserById' method is highlighted with a red box and labeled '1.'.

AppUser

1. GET /api/AppUser/GetApplicationUserById/{id}

PATCH /api/AppUser/UpdateApplicationUser

DELETE /api/AppUser/DeleteApplicationUser

POST /api/AppUser/AddChurchToApplicationUser

DELETE /api/AppUser/RemoveChurchFromApplicationUser

GET /api/AppUser/GetApplicationUserById/{id}

Parameters

2. Try it out

Name Description

id * required
string(\$uuid)
(path)

GET /api/AppUser/GetApplicationUserById/{id}

Parameters

Cancel

Name Description

id * required
string(\$uuid)
(path)

3. 91d461a3-e115-4b95-91d4-12a888c1a4ed

4. Execute

200

Response body

```
{
  "id": "91d461a3-e115-4b95-91d4-12a888c1a4ed",
  "userName": "TestUser1",
  "firstName": "First",
  "lastName": "LastName",
  "email": "user@example.com"
}
```

Response headers

```
content-type: application/json; charset=utf-8
date: Sat, 30 Apr 2022 22:48:01 GMT
server: Kestrel
x-firefox-spdy: h2
```

Description	<p>To get User Application By Id refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the App User section of the API Website 2. Under the Get function look for the “try it out” button on the right hand side 3. From there input the Id of the User you are trying to pull 4. Execute the function 5. A 200 ok response should appear with the information of the user who the id corresponds to.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserController.cs <p>Application Layer src → Core → Contracts → Identity src → Core → Features → ApplicationUser</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAppUserService.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence src → Infrastructure → ChurchPoint.Persistence → Services</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserService.cs



10.3.2 Update Application User

The screenshot shows a REST API documentation interface for the 'AppUser' resource. The 'UpdateApplicationUser' endpoint (PATCH /api/AppUser/UpdateApplicationUser) is highlighted with a red box and labeled '1.'.

Below the endpoint, the 'Try it out' button is also highlighted with a red box and labeled '2.'.

The 'Request body' section contains the following JSON example:

```
{  
  "applicationUserId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",  
  "firstName": "string",  
  "lastName": "string",  
  "phoneNumber": "string"  
}
```

Further down, the 'Request body' field is populated with the same JSON example, highlighted with a red box and labeled '3.'.

A large blue box highlights the 'Execute' button at the bottom, labeled '4.'.



Responses

```
Curl
curl -X 'PATCH' \
  'https://localhost:7252/api/AppUser/UpdateApplicationUser' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjUzNzQxNzIyMzg0LTQ5MzEtODc3ZC0xYTVjZTAwMDAifQ.' \
  -H 'Content-Type: application/json' \
  -d '{
    "applicationUserId": "91d461a3-e115-4b95-91d4-12a888c1a4ed",
    "firstName": "CSC",
    "lastName": "SeniorProject",
    "phoneNumber": "string"
  }'
```

Request URL

<https://localhost:7252/api/AppUser/UpdateApplicationUser>

Server response

Code	Details
204	Response headers date: Sat, 30 Apr 2022 22:59:22 GMT server: Kestrel x-firefox-spdy: h2

Responses

Code	Description	Links
204	Success	No links

Description	To update the application user refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the AppUser section of the API Website 2. Under the patch for the update application user there should be a “Try it out” button off to the right hand side 3. A text box should appear asking for the Id, first name, last name, and phone number of the user that needs to be updated 4. Execute the command 5. If done correctly there should be a 204 success response generated letting the user know that the update(patch) was successful.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. AppUserController.cs Application Layer



src → Core → Contracts → Identity
src → Core → Features → ApplicationUserCommand → DeleteAppUser

Then look at the following files:

1. IAppUserService.cs
2. DeleteAppUserCommand.cs
3. DeleteAppUserCommandHandler.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Services

Then look at the following files:

1. AppUserService.cs

10.3.3 Delete Application User

The screenshot shows a REST API documentation interface for the `AppUser` endpoint. The top section lists several methods:

- `GET /api/AppUser/GetApplicationUserById/{id}` (with a lock icon)
- `PATCH /api/AppUser/UpdateApplicationUser` (with a lock icon)
- `1 DELETE /api/AppUser/DeleteApplicationUser` (highlighted with a red border and numbered 1)
- `POST /api/AppUser/AddChurchToApplicationUser` (with a lock icon)
- `DELETE /api/AppUser/RemoveChurchFromApplicationUser` (with a lock icon)

Below this, the `DELETE /api/AppUser/DeleteApplicationUser` method is expanded:

- Parameters**: A table with one row:

Name	Description
<code>id</code>	<code>string(\$uuid)</code> (query)
- 2 Try it out**: A button with a red border and number 2.



Description	<p>To delete an application users refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the delete application user from the API Website 2. Under the delete function on the right, a “Try it out” button should appear 3. Once clicked, input the GUID of the user that needs to be deleted 4. Execute the command 5. A response should appear with either a 204 response indicating that the user was successfully deleted or a 404 Not Found response indicating that there was an error when deleting the user.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserController.cs <p>Application Layer src → Core → Contracts → Identity src → Core → Features → ApplicationUserCommand → DeleteAppUser</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAppUserService.cs 2. DeleteAppUserCommand.cs 3. DeleteAppUserCommandHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Services</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserService.cs



10.3.4 Add Church To Application User

The screenshot shows a sequence of five numbered steps for adding a church to an application user via a REST API.

- Step 1:** The top navigation bar lists several API endpoints:
 - GET /api/AppUser/GetApplicationUserById/{ id}
 - PATCH /api/AppUser/UpdateApplicationUser
 - DELETE /api/AppUser/DeleteApplicationUser
 - POST /api/AppUser/AddChurchToApplicationUser** (highlighted with a red box)
 - DELETE /api/AppUser/RemoveChurchFromApplicationUser
- Step 2:** The "Try it out" button for the POST endpoint is highlighted with a red box.
- Step 3:** The request body is filled with JSON data:

```
{ "appUserId": "5f48594-5712-4563-b0fc-2c9d0f6afac", "churchId": "5f48594-5712-4563-b0fc-2c9d0f6afac" }
```
- Step 4:** The "Execute" button is highlighted with a red box.
- Step 5:** The server response shows a 204 status code and the following response headers:

```
date: Fri, 13 May 2022 00:55:00 GMT
server: Microsoft-IIS/10.0
x-powered-by: ASP.NET
Content-Type: application/json; charset=utf-8
Content-Length: 0
```



Description	<p>To add a church from an application user refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the AppUser Section and click on AddChurchFromApplicationUser 2. Click on “Try it out” 3. Check request body dropdown to be “application/json” 4. Input the specified test input seen below 5. Press “Execute” 6. Response should be 204 No Content
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserController.cs <p>Application Layer src → Core → Contracts → Identity src → Core → Features → ApplicationUserCommand → AddChurchToAppUserId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAppUserService.cs 2. AddChurchToUserCommand.cs 3. AddChurchToUserCommandHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Services</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AppUserService.cs



10.3.5 Remove Church From Application User

The screenshot shows a REST API documentation interface for the `AppUser` resource. The `DELETE` method at `/api/AppUser/RemoveChurchFrom ApplicationUser` is highlighted with a red box and labeled '1'. A second red box highlights the 'Try it out' button next to the request URL. A third red box highlights the 'application/json' dropdown in the 'Request body' section. A fourth red box highlights the JSON request body. A fifth red box highlights the 'Execute' button at the bottom.

AppUser

1. `DELETE /api/AppUser/RemoveChurchFrom ApplicationUser`

2. Try it out

Parameters

No parameters

Request body application/json

Example Value | Schema

```
{ "appId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "churchId": "3fa85f64-5717-4562-b3fc-2c963f66afa6" }
```

3. application/json

4. { "appId": "3fa85f64-5717-4562-b3fc-2c963f66afa6", "churchId": "3fa85f64-5717-4562-b3fc-2c963f66afa6" }

5. Execute

Server response

Code	Details
204	Response headers date: Sat, 30 Apr 2022 23:37:32 GMT server: Kestrel

Responses

Code	Description	Links
------	-------------	-------

Description	To delete a church from an application user refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the AppUser Section and click on RemoveChurchFromApplicationUser 2. Click on “Try it out” 3. Check request body dropdown to be “application/json” 4. Input the specified test input seen below 5. Press “Execute” 6. Response should be 204 No Content
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. AppUserController.cs Application Layer src → Core → Contracts → Identity src → Core → Features → ApplicationUserCommand → RemoveChurchFromAppUser Then look at the following files: 1. IAppUserService.cs 2. RemoveChurchFromAppUserCommand.cs 3. RemoveChurchFromAppUserCommandHandler.cs Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Services Then look at the following files: 1. AppUserService.cs



10.4 Attendees API Functions

10.4.1 Get Attendee By Id

Attendees

1. GET /api/Attendees/GetAttendeeById/{id}

GET /api/Attendees/GetAllAttendeesByChurchId

GET /api/Attendees/GetAllAttendeesByEventId

GET /api/Attendees/GetAllAttendeesByMinistryServiceId

POST /api/Attendees/CreateAttendee

PATCH /api/Attendees/UpdateAttendee

DELETE /api/Attendees/DeleteAttendee

DELETE /api/Attendees/RemoveMinistryServiceFromAttendeeById

DELETE /api/Attendees/RemoveEventFromAttendeeById

2. Try it out

GET /api/Attendees/GetAttendeeById/{id}

Parameters

Name	Description
id <small>* required</small>	string(\$uuid) (path)

3. 3cb64d56-3906-4c6f-99c7-0a3af3bf679b

4. Execute



Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7252/api/Attendees/GetAttendeeById/3cb64d56-3906-4c6f-99c7-0a3af3bf679b' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NilsInR5cIi6IkpxVC09eyJzdWIiOiJUZXNWVXNlcjEiLCJqdGkiOiI2MmF1ZGU5OS1jMzp0LTQ5MzEt0Oc3ZC0xyYTViZTA'
```

Request URL

<https://localhost:7252/api/Attendees/GetAttendeeById/3cb64d56-3906-4c6f-99c7-0a3af3bf679b>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "attendeeId": "3cb64d56-3906-4c6f-99c7-0a3af3bf679b", "firstName": "Landon", "lastName": "Ashley", "address": "933 Reagan Extension , Virginia(VA), 24091", "phoneNumber": "+1 202-918-2132", "position": null, "email": "LandonAshley@email.com" }</pre> <p>5.</p> <p>Copy Download</p> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Sat, 30 Apr 2022 23:25:06 GMT server: Kestrel x-firefox-spdy: h2</pre>

Description	To get an attendee Id to refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the Attendees section of the Website API 2. Under getting attendee by Id there will be a “try it out” button on the right-hand side of the drop-down 3. Input the Id of the attendee that is needed 4. Execute the function 5. A 200 OK response should appear with all the information of the Attendee that corresponds to that Id.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. AttendeesController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features → Attendees → Query → GetAttendeeDetail Then look at the following files: 1. IAttendeeRepository.cs



- 2. AttendeeDetailVm.cs
- 3. GetAttendeeDetailQuery.cs
- 4. GetAttendeeDetailQueryHandler.cs

Infrastructure Layer
src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

- 1. AttendeeRepository.cs
- 2. BaseRepository.cs

10.4.2 Get Attendee By Church Id

Attendees

1. **GET** /api/Attendees/GetAllAttendeesByChurchId

2. Try it out

Other methods listed:

- GET /api/Attendees/GetAllAttendeesByEventId
- GET /api/Attendees/GetAllAttendeesByMinistryServiceId
- POST /api/Attendees/CreateAttendee
- PATCH /api/Attendees/UpdateAttendee
- DELETE /api/Attendees/DeleteAttendee
- DELETE /api/Attendees/RemoveMinistryServiceFromAttendeeById
- DELETE /api/Attendees/RemoveEventFromAttendeeById

GET /api/Attendees/GetAllAttendeesByChurchId

Parameters

2. Try it out

Name	Description
id	string(\$uuid) (query)



GET /api/Attendees/GetAllAttendeesByChurchId

Parameters

Name	Description
id string(\$uuid) (query)	3. id

4. Execute

Server response

Code Details

200 Response body

```
[  
  {  
    "attendeeId": "3cb64d56-3906-4c6f-99c7-0a3af3bf679b",  
    "firstName": "Landon",  
    "lastName": "Ashley",  
    "address": "933 Reagan Extension , Virginia(VA), 24091",  
    "phoneNumber": "+1 202-918-2132",  
    "position": null,  
    "email": "LandonAshley@email.com"  
  },  
  {  
    "attendeeId": "798082d0-d81e-4749-accf-25f17518d1b1",  
    "firstName": "Kennedy",  
    "lastName": "Scott",  
    "address": "9809 Hodgenville Rd Campbell, Kentucky, 427185",  
    "phoneNumber": "+1 239-677-3166",  
    "position": null,  
    "email": "amtsvorsteplers@hypoordip.live"  
  },  
  {  
    "attendeeId": "0a05b00d-94f1-49be-b101-b0e57758dc52",  
    "firstName": "Christopher",  
    "lastName": "Monroe",  
    "address": "119 S Cedar St Nokomis, Illinois, 62075",  
    "phoneNumber": "+1 582-500-2757",  
    "position": null,  
    "email": "flooded@silnmy.com"  
  }]
```

5.

Response headers

```
content-type: application/json; charset=utf-8  
date: Mon, 02 May 2022 00:52:05 GMT  
server: Kestrel
```

Responses

Description	To get an attendee by Church Id refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the get all attendees by church Id in the Website API 2. On the right side of the drop down click on “Try it out” 3. Enter the Id of the Church which list is needed
--------------------	--



	<p>4. Execute the function</p> <p>5. 200 OK Response with the list of attendees that correspond to the church Id.</p>
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ul style="list-style-type: none"> 1. AttendeesController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Attendees → Query → GetAllAttendeesByChurchId</p> <p>Then look at the following files:</p> <ul style="list-style-type: none"> 1. IAttendeeRepository.cs 2. AttendeeChurchListVm.cs 3. GetAllAttendeesByChurchIdQuery.cs 4. GetAllAttendeesByChurchIdQueryHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ul style="list-style-type: none"> 1. AttendeeRepository.cs 2. BaseRepository.cs



10.4.3 Get Attendee By Event Id

Attendees

GET	/api/Attendees/GetAttendeeById/{id}	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByChurchId	▼ 🔒
1.	GET /api/Attendees/GetAllAttendeesByEventId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByMinistryServiceId	▼ 🔒
POST	/api/Attendees/CreateAttendee	▼ 🔒
PATCH	/api/Attendees/UpdateAttendee	▼ 🔒
DELETE	/api/Attendees/DeleteAttendee	▼ 🔒
DELETE	/api/Attendees/RemoveMinistryServiceFromAttendeeById	▼ 🔒
DELETE	/api/Attendees/RemoveEventFromAttendeeById	▼ 🔒

1. GET /api/Attendees/GetAllAttendeesByEventId

Parameters

Name	Description
id	string(\$uuid) (query)

2. Try it out

2. Try it out

3. id

4. Execute



Server response

Code	Details
200	<p>Response body</p> <pre>[{ "attendeeId": "3cb64d56-3906-4c6f-99c7-0a3af3bf679b", "firstName": "Landon", "lastName": "Ashley", "address": "933 Reagan Extension , Virginia(VA), 24091", "phoneNumber": "+1 202-918-2132", "position": null, "email": "LandonAshley@email.com" }, { "attendeeId": "d6cccf27-4529-4d1f-acf1-68fce0810021", "firstName": "Harmony", "lastName": "Peterson", "address": "Po Box 1616 Gambier, Ohio(OH), 430223", "phoneNumber": "+1 582-999-9222", "position": null, "email": "miseryds@boranora.com" }, { "attendeeId": "b47fb59-f0f9-4380-b053-b398c23a564e", "firstName": "Luka", "lastName": "Perez", "address": "Po Box 384 Waverly Hall, Georgi, 31831", "phoneNumber": "+1 582-282-2941", "position": null, "email": "LukaPerez@email.com" }]</pre> <p>Copy Download</p> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Mon,02 May 2022 01:02:03 GMT server: Kestrel</pre>

Description	<p>To get an attendee by Event Id refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the get all attendees by Event Id in the Website API 2. On the right side of the drop down click on “Try it out” 3. Enter the Event Id that is trying to be reached 4. Execute the command 5. 200 OK with the list of attendees corresponding to the Event Id
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeesController.cs <p>Application Layer</p> <p>src → Core → Contracts → Persistence</p> <p>src → Core → Features→ Attendees→ Query→ GetAllAttendeesByEventId</p> <p>Then look at the following files:</p>



1. IAttendeeRepository.cs
2. AttendeeEventListVm.cs
3. GetAllAttendeesByEventIdQuery.cs
4. GetAllAttendeesByEventIdQueryHandler.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. AttendeeRepository.cs
2. BaseRepository.cs

10.4.4 Get Attendee By Ministry Service Id

Attendees

GET	/api/Attendees/GetAttendeeById/{id}	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByChurchId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByEventId	▼ 🔒
1. GET	/api/Attendees/GetAllAttendeesByMinistryServiceId	▼ 🔒
POST	/api/Attendees/CreateAttendee	▼ 🔒
PATCH	/api/Attendees/UpdateAttendee	▼ 🔒
DELETE	/api/Attendees/DeleteAttendee	▼ 🔒
DELETE	/api/Attendees/RemoveMinistryServiceFromAttendeeById	▼ 🔒
DELETE	/api/Attendees/RemoveEventFromAttendeeById	▼ 🔒

GET /api/Attendees/GetAllAttendeesByMinistryServiceId

Parameters

2. Try it out

Name	Description
id string(\$uuid) (query)	<input type="text" value="id"/>



GET /api/Attendees/GetAllAttendeesByMinistryServiceId

Parameters

Name	Description
id string(\$uuid) (query)	3. id

4. Execute

Server response

Code Details

200 Response body

```
[  
  {  
    "attendeeId": "0a05b00d-94f1-49be-b101-b0e57758dc52",  
    "firstName": "Christopher",  
    "lastName": "Monroe",  
    "address": "119 S Cedar St Nokomis, Illinois, 62075",  
    "phoneNumber": "+1 582-500-2757",  
    "position": null,  
    "email": "flooded@silnmy.com"  
  },  
  {  
    "attendeeId": "b47fb59-f0f9-4380-b053-b398c23a564e",  
    "firstName": "Luka",  
    "lastName": "Perez",  
    "address": "Po Box 384 Waverly Hall, Georgi, 31831",  
    "phoneNumber": "+1 582-282-2941",  
    "position": null,  
    "email": "LukaPerez@email.com"  
  }]
```

5.  

Response headers

```
content-type: application/json; charset=utf-8  
date: Mon, 02 May 2022 01:13:42 GMT  
server: Kestrel
```

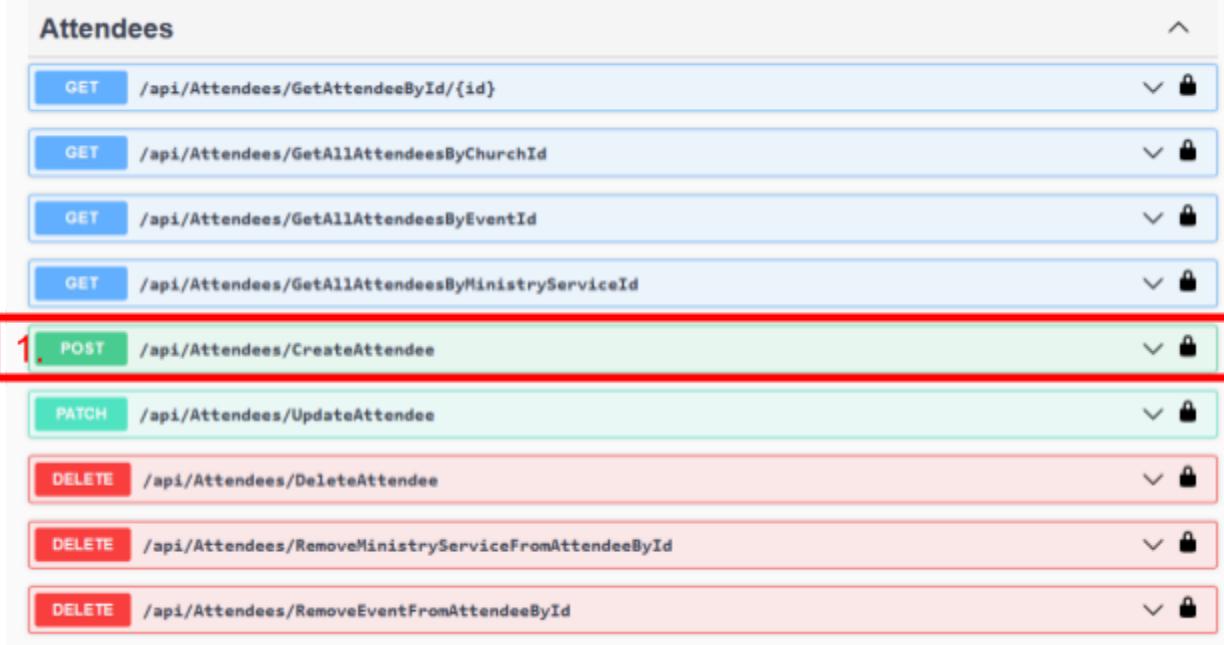
Responses

Description	To get an attendee by Ministry Service Id refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the get all attendees by ministry service id section under the Attendees tab in the Website API 2. On the right side of the drop down click on “Try it out” 3. Enter the Ministry Service Id 4. Execute the function 5. A 200 Ok Response should ensue with the list of the attendees that correspond to that Ministry Service Id.
--------------------	---



<p>List of Code Files</p> <p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeesController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Attendees → Query → GetAllAttendeesByMinistryServiceId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAttendeeRepository.cs 2. AttendeeMinistryServiceListVm.cs 3. GetAllAttendeesByMinistryServiceIdQueryHandler.cs 4. GetAllAttendeesByMinistryServiceIdQuery.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeeRepository.cs 2. BaseRepository.cs
--

10.4.5 Create Attendee User



The screenshot shows a list of API endpoints for the 'Attendees' controller. The 'POST /api/Attendees/CreateAttendee' endpoint is highlighted with a red box.

Attendees		
GET	/api/Attendees/GetAttendeeById/{id}	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByChurchId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByEventId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByMinistryServiceId	▼ 🔒
1. POST	/api/Attendees/CreateAttendee	▼ 🔒
PATCH	/api/Attendees/UpdateAttendee	▼ 🔒
DELETE	/api/Attendees/DeleteAttendee	▼ 🔒
DELETE	/api/Attendees/RemoveMinistryServiceFromAttendeeById	▼ 🔒
DELETE	/api/Attendees/RemoveEventFromAttendeeById	▼ 🔒



POST /api/Attendees/CreateAttendee

Parameters

No parameters

Request body

application/json

Example Value | Schema

```
{  
    "firstName": "string",  
    "lastName": "string",  
    "email": "string",  
    "address": "string",  
    "phoneNumber": "string",  
    "churchId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"  
}
```

POST /api/Attendees/CreateAttendee

Parameters

No parameters

Request body

application/json

```
{  
    "firstName": "string",  
    "lastName": "string",  
    "email": "string",  
    "address": "string",  
    "phoneNumber": "string",  
    "churchId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"  
}
```

5. Execute



Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7252/api/Attendees/CreateAttendee' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJUZXN0VXNlciIiImp0aSI6Ijh1NzkyYzkzMWNNTMtnDdINy1h \
  -H 'Content-Type: application/json' \
  -d '{
    "firstName": "Another",
    "lastName": "One",
    "email": "djkhaled@gmail.com",
    "address": "123 Major Key Alert Way",
    "phoneNumber": "999-888-1234",
    "churchId": "1250a558-7335-4a8a-8083-d3b178e3137b"
  }'
```

Request URL

<https://localhost:7252/api/Attendees/CreateAttendee>

Server response

Code	Details
201	<p>Response body</p> <p>6. "2d0215f9-3ac0-44f8-4cd6-08da2bda8b01"</p>

Download

Response Headers

```
content-type: application/json; charset=utf-8
date: Mon, 02 May 2022 01:24:46 GMT
server: Kestrel
```

Description	<p>To create an attendee refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the Create Attendee section under Attendees in the Website API 2. On the right side of the drop down click on “Try it out” 3. Make sure the parameter drop down is set to “application/json” 4. In the text box input the information to create a new attendee inside the quotes 5. Execute the function 6. A 201 Created should be executed, giving the new Guid of the attendee.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeesController.cs <p>Application Layer</p> <p>src → Core → Contracts → Persistence</p>



src → Core → Features → Attendees → Command → CreateAttendee

Then look at the following files:

1. IAttendeeRepository.cs
2. CreateAttendeeCommand.cs
3. CreateAttendeeCommandHandler.cs
4. CreateAttendeeCommandValidator.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. AttendeeRepository.cs
2. BaseRepository.cs

10.4.6 Update Attendee User

The screenshot shows a list of API endpoints for the 'Attendees' endpoint. The endpoints are:

- GET /api/Attendees/GetAttendeeById/{id}
- GET /api/Attendees/GetAllAttendeesByChurchId
- GET /api/Attendees/GetAllAttendeesByEventId
- GET /api/Attendees/GetAllAttendeesByMinistryServiceId
- POST /api/Attendees/CreateAttendee
- PATCH /api/Attendees/UpdateAttendee** (highlighted with a red box)
- DELETE /api/Attendees/DeleteAttendee
- DELETE /api/Attendees/RemoveMinistryServiceFromAttendeeById
- DELETE /api/Attendees/RemoveEventFromAttendeeById

Below the list, there is a detailed view for the highlighted PATCH endpoint:

PATCH /api/Attendees/UpdateAttendee

Parameters

No parameters

Try it out (highlighted with a red box)



PATCH /api/Attendees/UpdateAttendee

Parameters

No parameters

Request body

3 application/json

4 {
 "attendeeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
 "firstName": "string",
 "lastName": "string",
 "phoneNumber": "string",
 "address": "string"
}

5. Execute

Responses

Curl

```
curl -X 'PATCH' \
  'https://localhost:7252/api/Attendees/UpdateAttendee' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJzdWIiOiJUZXRWVXNlcilisImp0aSI6IjhINzkyYzkzLWM0NTMhNDdiNy1h
  -H 'Content-Type: application/json' \
  -d '{
    "attendeeId": "2d0215f9-3ac0-44f8-4cd6-08da2bda8b01",
    "firstName": "string",
    "lastName": "string",
    "phoneNumber": "string",
    "address": "345 The One Drive"
}'
```

Request URL

<https://localhost:7252/api/Attendees/UpdateAttendee>

Server response

Code	Details
204	Response headers
6.	<pre>date: Mon, 02 May 2022 01:38:20 GMT server: Kestrel</pre>

Description	<p>To Update an attendee, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the Update Attendee section under the Attendees in the Website API 2. On the right side of the drop down click on “Try it out” 3. Make sure the parameter drop down is set to “application/json” 4. Input the data necessary for the update 5. Execute the function 6. A 204 No Content should be developed indicating the attendee was updated.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeesController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Attendees → Command → UpdateAttendee</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAttendeeRepository.cs 2. UpdateAttendeeCommand.cs 3. UpdateAttendeeCommandHandler.cs 4. UpdateAttendeeCommandValidator.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeeRepository.cs 2. BaseRepository.cs



10.4.7 Delete Attendee User

Attendees

GET	/api/Attendees/GetAttendeeById/{id}	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByChurchId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByEventId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByMinistryServiceId	▼ 🔒
POST	/api/Attendees/CreateAttendee	▼ 🔒
PATCH	/api/Attendees/UpdateAttendee	▼ 🔒
1. DELETE	/api/Attendees/DeleteAttendee	▼ 🔒
DELETE	/api/Attendees/RemoveMinistryServiceFromAttendeeById	▼ 🔒
DELETE	/api/Attendees/RemoveEventFromAttendeeById	▼ 🔒

DELETE /api/Attendees/DeleteAttendee

Parameters

Name	Description
id	string(\$uuid) (query)

2. Try it out

DELETE /api/Attendees/DeleteAttendee

Parameters

Name	Description
id	string(\$uuid) (query)

3. id

4. Execute



Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7252/api/Attendees/DeleteAttendee?id=2d0215f9-3ac0-44f8-4cd6-08da2bda8b01' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJUZXN0VXNlcjIiImp0aS16IjA3MWDkZjAwLWRkOWM1NGNhNy1h
  d...'
```

Request URL

<https://localhost:7252/api/Attendees/DeleteAttendee?id=2d0215f9-3ac0-44f8-4cd6-08da2bda8b01>

Server response

Code	Details
204	Response headers 5. date: Mon, 02 May 2022 01:48:53 GMT server: Kestrel

Description	<p>To delete an attendee, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the delete an attendee section under the Attendee in the Website API 2. On the right side of the drop down click on “Try it out” 3. Enter the Id of the Attendee that needs to be deleted 4. Execute the function 5. A 204 No Content should be in the response indicating that the Attendee has been deleted.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeesController.cs <p>Application Layer</p> <p>src → Core → Contracts → Persistence src → Core → Features → Attendees → Command → DeleteAttendee</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAttendeeRepository.cs 2. DeleteAttendeeCommand.cs 3. DeleteAttendeeCommandHandler.cs



Infrastructure Layer
src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. AttendeeRepository.cs
2. BaseRepository.cs

10.4.8 Remove Ministry Service From Attendee By Id

Attendees

GET	/api/Attendees/GetAttendeeById/{id}	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByChurchId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByEventId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByMinistryServiceId	▼ 🔒
POST	/api/Attendees/CreateAttendee	▼ 🔒
PATCH	/api/Attendees/UpdateAttendee	▼ 🔒
DELETE	/api/Attendees/DeleteAttendee	▼ 🔒
1. DELETE	/api/Attendees/RemoveMinistryServiceFromAttendeeById	▼ 🔒
DELETE	/api/Attendees/RemoveEventFromAttendeeById	▼ 🔒

DELETE /api/Attendees
/RemoveMinistryServiceFromAttendeeById

Parameters

No parameters

2. Try it out



DELETE /api/Attendees
/RemoveMinistryServiceFromAttendeeById

Parameters Cancel

No parameters

Request body 3. application/json ▾

```
{
  "attendeeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "ministryServiceId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

4.

5. Execute

Description	To Remove a ministry service Id from attendee by Id, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none">1. Navigate to the Remove Ministry Service by Attendee Id section under Attendee in the Website API2. On the right side of the drop down click on “Try it out”3. Make sure the parameter drop down is set to “application/json”
-------------	---

	<ol style="list-style-type: none"> 4. Input the correct Ministry Service Id and Attendee Id 5. Execute the function 6. A 204 No Content Response should be generated allowing the user to know that the ministry service was deleted from the corresponding attendee Id.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeesController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Attendees → Command → RemoveMinistryServiceFromAttendeeId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAttendeeRepository.cs 2. RemoveMinistryServiceFromAttendeeIdCommand.cs 3. RemoveMinistryServiceFromAttendeeIdCommandHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeeRepository.cs 2. BaseRepository.cs



10.4.9 Remove Event From Attendee By Id

Attendees

GET	/api/Attendees/GetAttendeeById/{id}	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByChurchId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByEventId	▼ 🔒
GET	/api/Attendees/GetAllAttendeesByMinistryServiceId	▼ 🔒
POST	/api/Attendees/CreateAttendee	▼ 🔒
PATCH	/api/Attendees/UpdateAttendee	▼ 🔒
DELETE	/api/Attendees/DeleteAttendee	▼ 🔒
DELETE	/api/Attendees/RemoveMinistryServiceFromAttendeeById	▼ 🔒
1. DELETE	/api/Attendees/RemoveEventFromAttendeeById	▼ 🔒

1. DELETE /api/Attendees/RemoveEventFromAttendeeById

Parameters

No parameters

2. Try it out



DELETE /api/Attendees/RemoveEventFromAttendeeById

Parameters

No parameters

Request body

3. application/json

4. {
 "attendeeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
 "eventId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
 }

Execute

Responses

Curl

```
curl -X "DELETE" \
  "https://localhost:7252/api/attendees/RemoveEventFromAttendeeById" \
  -H "accept: text/plain" \
  -H "Authorization: bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6ImlhdmlldmVycyIsImVhZG15IjoiMSIsInByb2ZpbGUiOiJodHRwczovL3d3dy5nb2RpLmNvbS8xMDAifQ.eyJpc3MiOiJodHRwczovL3d3dy5nb2RpLmNvbS8xMDAiLCJ0eXAiOiJKV1QiLCJ4eXAiOiJKV1QiLCJzY29wZWQiOiJodHRwczovL3d3dy5nb2RpLmNvbS8xMDAifQ" \
  -H "Content-Type: application/json" \
  -d '{ "attendeeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6" , "eventId": "3fa85f64-5717-4562-b3fc-2c963f66afa6" }'
```

Request URL

<https://localhost:7252/api/Attendees/RemoveEventFromAttendeeById>

Server response

Code	Details
204	6. Response headers <code>date: Mon, 02 May 2022 01:15:57 GMT</code> <code>server: Kestrel</code> <code>x-aspnet-version: 4.0</code>

Description	To Remove Event From Attendee by Id, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the Remove Event From Attendee by Id section under Attendee in the Website API 2. On the right side of the drop down click on “Try it out” 3. Make sure the parameter drop down is set to “application/json”
--------------------	--



	<ol style="list-style-type: none"> 4. Input the correct Attendee Id and Event Id 5. Execute the Function 6. A 204 No content Code should be the response indicating that the Event has been removed from attendee by Id.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeesController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Attendees → Command → RemoveEventFromAttendeeId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IAttendeeRepository.cs 2. RemoveEventFromAttendeeIdCommand.cs 3. RemoveEventFromAttendeeIdCommandHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. AttendeeRepository.cs 2. BaseRepository.cs

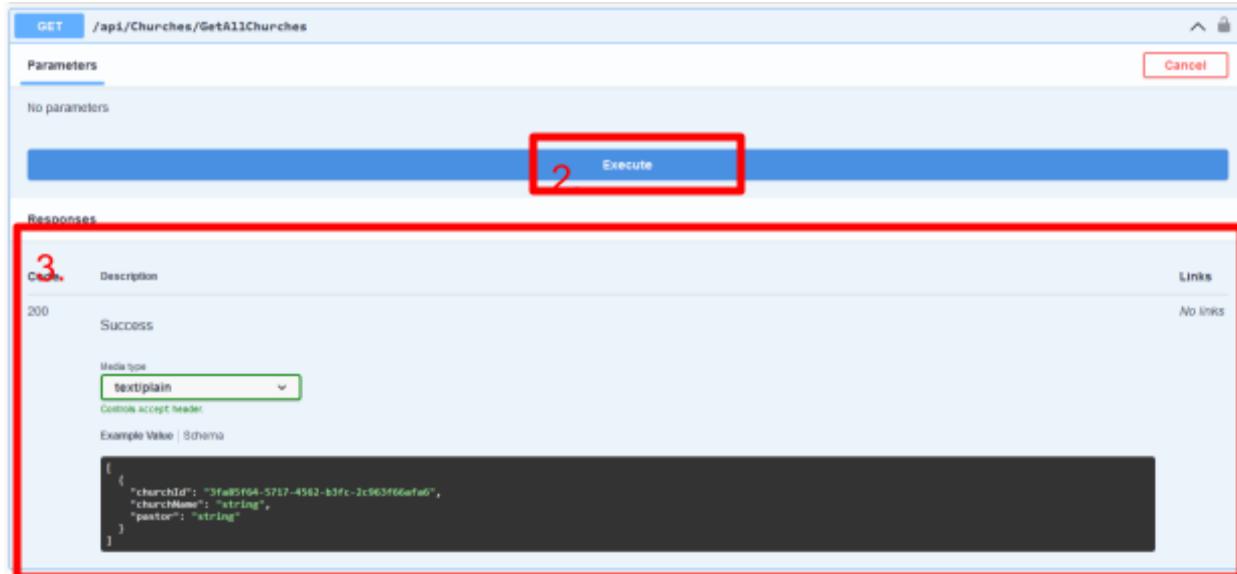
10.5 Churches API Functions

10.5.1 Get All Churches

The screenshot shows a list of API endpoints for the 'Churches' endpoint. The first endpoint, 'GET /api/Churches/GetAllChurches', is highlighted with a red box. The other endpoints listed are:

- GET /api/Churches/GetChurchById/{id}
- GET /api/Churches/GetAllChurchesByApplicationUserId
- GET /api/Churches/GetAllChurchesByAttendeeId
- POST /api/Churches/CreateChurch
- PATCH /api/Churches/UpdateChurch
- DELETE /api/Churches/DeleteChurch
- POST /api/Churches/AddAttendeeToChurchById
- DELETE /api/Churches/RemoveAttendeeFromChurchById





Description	To get all Churches refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the Get All Churches section under Churches in the Website API 2. On the right side of the drop down click on “Try it out” and click “Execute” 3. Make sure the parameter drop down is set to “text/plain”
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. ChurchesController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features → Churches → Query → GetChurchList Then look at the following files: 1. IChurchRepository.cs 2. ChurchListVm.cs 3. GetChurchesListQuery.cs 4. GetChurchesListQueryHandler.cs Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories Then look at the following files:

	<ol style="list-style-type: none"> 1. ChurchRepository.cs 2. BaseRepository.cs
--	--

10.5.2 Get Church By Id

The screenshot displays the Church API documentation. The top section shows a list of endpoints under the 'Churches' category:

- GET /api/Churches/GetAllChurches
- 1. GET /api/Churches/GetChurchById/{id}** (highlighted with a red box)
- GET /api/Churches/GetAllChurchesByApplicationUserId
- GET /api/Churches/GetAllChurchesByAttendeeId
- POST /api/Churches/CreateChurch
- PATCH /api/Churches/UpdateChurch
- DELETE /api/Churches/DeleteChurch
- POST /api/Churches/AddAttendeeToChurchById
- DELETE /api/Churches/RemoveAttendeeFromChurchById

The bottom section provides details for the 'GetChurchById' endpoint:

GET /api/Churches/GetChurchById/{id}

Parameters

Name	Description
id *	string The church id. (path)

Responses

Code	Description	Links
200	Success	No links

Content Type: text/plain

Example Value: Schema

```
{
  "churchId": "3fa85f04-5717-4542-b3fc-3c963700ef66",
  "churchName": "string",
  "phoneNumber": "string",
  "address": "string",
  "denomination": "string",
  "pastor": "string"
}
```

Description	To get Church information by ID, refer to the figures above for a sequential visual guide starting with label 1 <ol style="list-style-type: none"> 1. Navigate to Get Churches by ID in the website API 2. Click “Try it out” on the top right side of the GetChurchesById UI 3. Input id
--------------------	--



<p>List of Code Files</p> <p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. ChurchesController.cs <p>Application Layer</p> <p>src → Core → Contracts → Persistence</p> <p>src → Core → Features → Churches → Query → GetChurchDetail</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IChurchRepository.cs 2. ChurchDetailVm.cs 3. GetChurchDetailQuery.cs 4. GetChurchDetailQueryHandler.cs <p>Infrastructure Layer</p> <p>src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. ChurchRepository.cs 2. BaseRepository.cs
--

10.5.3 Get All Churches By Application User Id

The screenshot shows a list of API endpoints for the 'Churches' controller. The endpoints are:

- GET /api/Churches/GetAllChurches
- GET /api/Churches/GetChurchById/{Id}
- 1. GET /api/Churches/GetAllChurchesByApplicationUserId
- GET /api/Churches/GetAllChurchesByAttendeeId
- POST /api/Churches/CreateChurch
- PATCH /api/Churches/UpdateChurch
- DELETE /api/Churches/DeleteChurch
- POST /api/Churches/AddAttendeeToChurchById
- DELETE /api/Churches/RemoveAttendeeFromChurchById

The third endpoint, '1. GET /api/Churches/GetAllChurchesByApplicationUserId', is highlighted with a red box.



GET /api/Churches/GetAllChurchesByApplicationUserId

Parameters

Name	Description
id	string (query)

Responses

Code	Description	Links
200	SUCCESS	No links

Media type: text/plain

```
[{"churchName": "string", "churchId": "3f8e845217-4562-b3fc-2c963f66afab", "denomination": "string", "address": "string", "pastor": "string", "phoneNumber": "string"}]
```

Description	<p>To get Church information by Application User ID, refer to the figures above for a sequential visual guide starting with label 1</p> <ol style="list-style-type: none"> 1. Navigate to Get All Churches By Application User ID 2. Click “Try it out” on the top right corner. 3. Input Application User ID in id text field 4. Click Execute.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. ChurchesController.cs <p>Application Layer</p> <p>src → Core → Contracts → Persistence</p> <p>src → Core → Features → Churches → Query → GetAllChurchesByApplicationUserId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IChurchRepository.cs 2. AppUserChurchesVm.cs 3. GetAppUserChurchesQuery.cs 4. GetAppUserChurchesQueryHandler.cs



Infrastructure Layer
src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. ChurchRepository.cs
2. BaseRepository.cs

10.5.4 Get All Churches By Attendee User Id

The screenshot shows the 'Churches' endpoint in a Swagger UI. The top section lists several GET methods:

- GET /api/Churches/GetAllChurches
- GET /api/Churches/GetChurchById/{id}
- GET /api/Churches/GetAllChurchesByApplicationUserId
- 1. GET /api/Churches/GetAllChurchesByAttendeeId

The method at index 1 is highlighted with a red box and labeled '1'. Below this, there are other methods: POST /api/Churches/CreateChurch, PATCH /api/Churches/UpdateChurch, DELETE /api/Churches/DeleteChurch, POST /api/Churches/AddAttendeeToChurchById, and DELETE /api/Churches/RemoveAttendeeFromChurchById.

The bottom section provides detailed information for the highlighted method (index 1):

- Method:** GET /api/Churches/GetAllChurchesByAttendeeId
- Parameters:** A table with a single row for 'id':
 - Name: id
 - Description: string (0)
 - Value: 3 (highlighted with a red box)
- Responses:** A table for status code 200:
 - Code:** 200
 - Description:** Success
 - Media type:** text/plain (highlighted with a red box)
 - Example Value:** Schema (shown as a JSON object)

```
[{"churchName": "string", "churchId": "3fdd564-5737-4682-bdfc-2e963f66ef6", "denomination": "string", "address": "string", "postcode": "string", "phonenumber": "string"}]
```
 - Links:** No links



Parameters

Name	Description
id string(S guid) (query)	4196fe9-4f2e-4ff5-924a-2696a1a6b742

Execute **Cancel** **Clear**

Responses

Curl

```
curl -X GET \
  https://localhost:7252/api/Churches/GetAllChurchesByAttendeeId/4196fe9-4f2e-4ff5-924a-2696a1a6b742 \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6ImltYWdlcyBhbmVzIiwiaWF0IjoxNTE2MjM0MTIwLCJleHAiOjE2MTEyMjM0MTAsImF1ZCI6ImFkbWluIiwic2lnIjoiZW1haWwiLCJpYXQiOjE2MTEyMjM0MTAsInN1YiI6ImFkbWluIiwidmFsdWUiOiJsb2dpbiIsImV4cCI6MTIwMjIyMjM0NA==
```

Request URL

<https://localhost:7252/api/Churches/GetAllChurchesByAttendeeId/4196fe9-4f2e-4ff5-924a-2696a1a6b742>

Server response

Code	Details
200	<p>Response body</p> <pre>{ "churches": [{ "churchName": "Inbrace Gospel Church", "churchId": "4196fe9-4f2e-4ff5-924a-2696a1a6b742", "denomination": "Methodist General Conference", "address": "30th & Utica Rd, New York, NY 10001", "pastor": "James Turner", "phoneNumber": "312-234-2324" }, { "churchName": "Arise Church", "churchId": "e6c1d04-f8e6-4f45-8ff7-00400be30f", "denomination": "Association of Free Lutheran Congregations", "address": "1000 Queen St. Hayes, North Carolina, 28001", "pastor": "Alan White", "phoneNumber": "704-333-4400" }, { "churchName": "Celebrate Church", "churchId": "3c19-05-4732-4853-73f9fb61cf", "denomination": "None", "address": "8000 Stark Ave, Youngstown, Ohio, 34466", "pastor": "John Stevens", "phoneNumber": "330-977-9922" }, { "churchName": "West Hill Refuge Church", "churchId": "4e3e320e-0a00-4000-a000-000000000000", "denomination": "Adventist, Pentecostal, etc." }] }</pre> <p>5. Download</p> <p>Response Headers</p> <p>content-type: application/json; charset=utf-8 date: Sat, 07 May 2022 20:20:13 GMT</p>

Description	To get all Church information by Attendee ID, refer to the figures above for a sequential visual guide starting with label 1 <ol style="list-style-type: none"> 1. Navigate to Get All Churches by Attendee ID 2. Click Try it Out 3. input Attendee ID 4. Click Execute 5. A 200 Ok Response should be generated with a list of all churches
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. ChurchesController.cs Application Layer



src → Core → Contracts → Persistence
src → Core → Features → Churches → Query → GetAllChurchesByAttendeeId

Then look at the following files:

1. IChurchRepository.cs
2. AttendeeChurchesVm.cs
3. GetAllChurchesByAttendeeHandler.cs
4. GetAllChurchesByAttendeeQuery.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. ChurchRepository.cs
2. BaseRepository.cs

10.5.5 Create Church User

The screenshot shows a list of API endpoints for the 'Churches' endpoint. The methods listed are:

- POST /api/Churches/CreateChurch (highlighted with a red box)
- PATCH /api/Churches/UpdateChurch
- DELETE /api/Churches/DeleteChurch
- POST /api/Churches/AddAttendeeToChurchById
- DELETE /api/Churches/RemoveAttendeeFromChurchById

The screenshot shows the details for the POST /api/Churches/CreateChurch method. It includes:

- Method: POST /api/Churches/CreateChurch
- Parameters: No parameters
- Request body: application/json
- Example Value | Schema:

```
{"churchName": "string", "address": "string", "pastor": "string", "phonenumbers": "string", "denomination": "string"}
```
- A red box highlights the 'Try it out' button.



POST /api/Churches/CreateChurch

Parameters

No parameters

Request body

application/json

3.

```
{
  "churchName": "The Hills Church",
  "address": "123 Torrada Way, California 12300",
  "pastor": "John Doe",
  "phone": "555-555-5555",
  "denomination": "Baptist"
}
```

4. Execute

Request URL: <https://localhost:7250/api/Churches/CreateChurch>

Server response

Code Details

201 Response body

```
"5e0d154-f211-49c6-6405-000000000000"
```

5. Response headers

```
content-type: application/json; charset=utf-8
date: Sun, 08 May 2022 01:48:06 GMT
server: Kestrel
x-powered-by: .NET
```

Responses

Description	To Create Church, refer to the figures above for a sequential visual guide starting with label 1 <ol style="list-style-type: none"> 1. Navigate to Create Church module 2. Click Try it out 3. Input details about the church on the text box 4. Click Execute 5. 201 code and unique ChurchId will be returned.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. ChurchesController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features → Churches → Command → CreateChurch



Then look at the following files:

1. IChurchRepository.cs
2. CreateChurchCommand.cs
3. CreateChurchCommandHandler.cs
4. CreateChurchCommandValidator.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. ChurchRepository.cs
2. BaseRepository.cs

10.5.6 Update Church User

The screenshot shows a list of API endpoints for the 'Churches' endpoint. The 'PATCH /api/Churches/UpdateChurch' endpoint is highlighted with a red box. Below it, a detailed view of the same endpoint is shown, also with a red box around the 'Try it out' button.

Churches

- GET /api/Churches/GetAllChurches
- GET /api/Churches/GetChurchById/{id}
- GET /api/Churches/GetAllChurchesByApplicationUserId
- GET /api/Churches/GetAllChurchesByAttendeeId
- POST /api/Churches/CreateChurch**
- 1. PATCH /api/Churches/UpdateChurch**
- DELETE /api/Churches/DeleteChurch
- POST /api/Churches/AddAttendeeToChurchById
- DELETE /api/Churches/RemoveAttendeeFromChurchById

PATCH /api/Churches/UpdateChurch

Parameters

No parameters

Request body

application/json

Example Value | Schema

```
[{"churchId": "f1eaf16a-5017-4560-b0fc-3c9d0fcaefaf", "churchName": "string", "address": "string", "city": "string", "phoneNumber": "string", "denomination": "string"}]
```

2 Try it out



PATCH /api/Churches/UpdateChurch

Parameters
No parameters

Request body (application/json)

```
{
  "churchId": "mebaeb4-f221-48bc-a040-0d4d304cc0ce",
  "churchName": "Calvary",
  "address": "725 Ruth Way, Washington",
  "pastor": "Jeremy Lin",
  "phoneNumber": "988-241-8777",
  "denomination": "Christian"
}
```

Responses

Curl

```
curl -X 'PATCH' \
  'https://localhost:7251/api/Churches/UpdateChurch' \
  -H 'accept: */*' \
  -H 'Authorization: bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJodHRwczovL2xvY2FsaG9zdC5jb20iLCJ0eXAiOiJKV1QiLCJ4bWxuczI6ImFjY2VudCIsImlhdCI6MTYyNjMwOTQ4MSwiZXhwIjoxNjI2Mzg5MDkxfQ.eyJpc3MiOiJodHRwczovL2xvY2FsaG9zdC5jb20iLCJ0eXAiOiJKV1QiLCJ4bWxuczI6ImFjY2VudCIsImlhdCI6MTYyNjMwOTQ4MSwiZXhwIjoxNjI2Mzg5MDkxfQ'
  -H 'Content-type: application/json' \
  -d '{
    "churchId": "mebaeb4-f221-48bc-a040-0d4d304cc0ce",
    "churchName": "Calvary",
    "address": "725 Ruth Way, Washington",
    "pastor": "Jeremy Lin",
    "phoneNumber": "988-241-8777",
    "denomination": "Christian"
  }'
```

Request URL

https://localhost:7251/api/Churches/UpdateChurch

Server response

Code	Details
204	Response headers <code>date: Sun, 08 May 2023 01:51:47 GMT</code> <code>server: Kestrel</code> <code>x-frame-options: No</code>
Responses	

Description	To Update Church details, refer to the figures above for a sequential visual guide starting with label 1 <ol style="list-style-type: none"> 1. Navigate to the Update Church module 2. Click “Try it out” 3. Input the Church ID as well as the details needed to be updated. <ol style="list-style-type: none"> a. If details are the same, it must be re-entered. 4. Click execute. 5. 204 return success
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers



	<p>API Layer Then look at the following files: 1. ChurchesController.cs</p> <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Churches → Command → UpdateChurch</p> <p>Then look at the following files: 1. IChurchRepository.cs 2. UpdateChurchCommand.cs 3. UpdateChurchCommandHandler.cs 4. UpdateChurchCommandValidator.cs</p> <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files: 1. ChurchRepository.cs 2. BaseRepository.cs</p>
--	---

10.5.7 Delete Church User

Churches		
GET	/api/Churches/GetAllChurches	▼ 🔒
GET	/api/Churches/GetChurchById/{id}	▼ 🔒
GET	/api/Churches/GetAllChurchesByApplicationUserId	▼ 🔒
GET	/api/Churches/GetAllChurchesByAttendeeId	▼ 🔒
POST	/api/Churches/CreateChurch	▼ 🔒
PATCH	/api/Churches/UpdateChurch	▼ 🔒
1. DELETE	/api/Churches/DeleteChurch	▼ 🔒
POST	/api/Churches/AddAttendeeToChurchById	▼ 🔒
DELETE	/api/Churches/RemoveAttendeeFromChurchById	▼ 🔒



DELETE /api/Churches/DeleteChurch

Parameters

Name	Description
id string(Guid) (query)	id

Responses

Code	Description	Links
204	Success	No links

Try it out 2

DELETE /api/Churches/DeleteChurch

Parameters

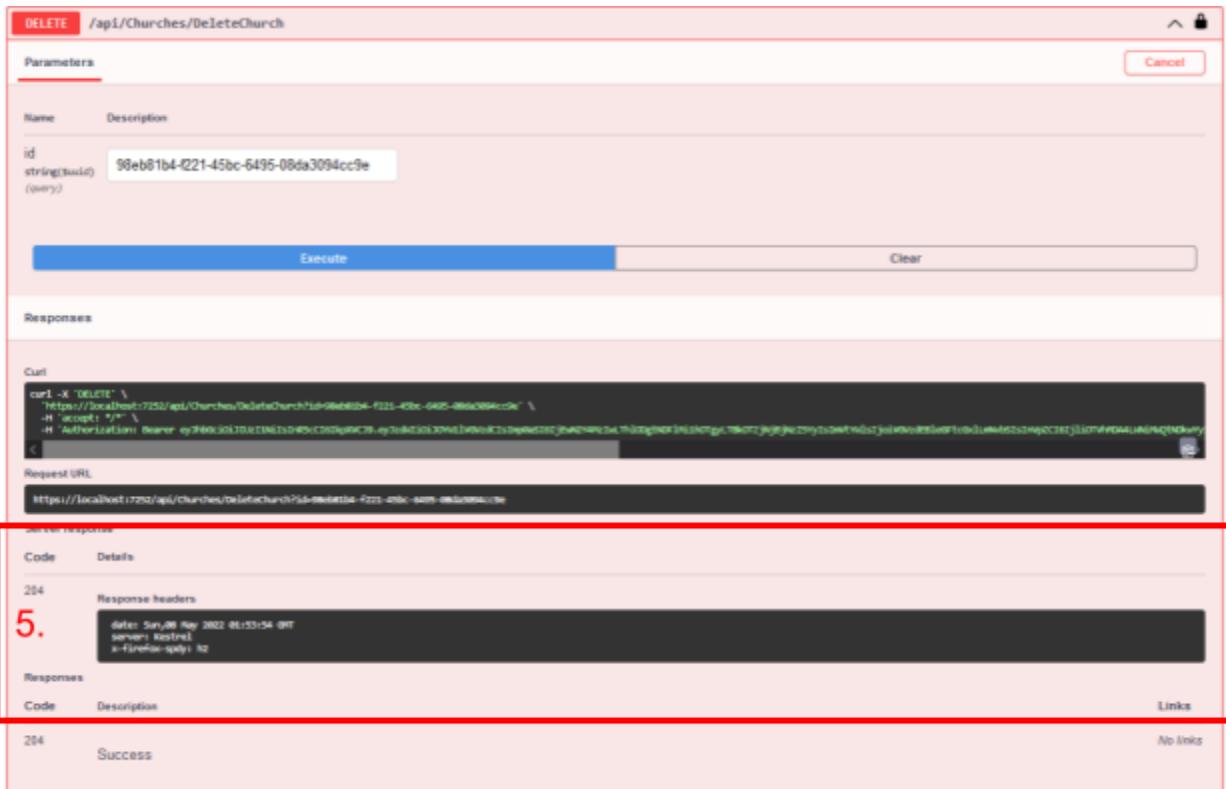
Name	Description
id string(Guid) (query)	3 98eb81b4-f221-45bc-6495-08da3094cc9e

Responses

Code	Description	Links
204	Success	No links

Cancel 4 Execute





Description	<p>To Delete Church details, refer to the figures above for a sequential visual guide starting with label 1</p> <ol style="list-style-type: none"> 1. Navigate to the Delete Church module 2. Click “Try it out” 3. Input ChurchId 4. click execute 5. Receive 204 Code Success
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. ChurchesController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Churches → Command → DeleteChurch</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IChurchRepository.cs



2. DeleteChurchCommand.cs
 3. DeleteChurchCommandHandler.cs

Infrastructure Layer
 src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. ChurchRepository.cs
 2. BaseRepository.cs

10.5.8 Add Attendee To Church By Id

Churches

GET	/api/Churches/GetAllChurches	▼	🔒
GET	/api/Churches/GetChurchById/{id}	▼	🔒
GET	/api/Churches/GetAllChurchesByApplicationUserId	▼	🔒
GET	/api/Churches/GetAllChurchesByAttendeeId	▼	🔒
POST	/api/Churches/CreateChurch	▼	🔒
PATCH	/api/Churches/UpdateChurch	▼	🔒
DELETE	/api/Churches/DeleteChurch	▼	🔒
1.	POST /api/Churches/AddAttendeeToChurchById	▼	🔒
DELETE	/api/Churches/RemoveAttendeeFromChurchById	▼	🔒

1. POST /api/Churches/AddAttendeeToChurchById

2. Try it out

Parameters

No parameters

Request body

application/json

Example Value | Schema

```
[{"attendeeId": "5fa6f664-5717-45d2-b2fc-3c46f66aa7a6", "churchId": "5fa6f664-5717-45d2-b2fc-3c46f66aa7a6"}]
```

Responses

Code	Description	Links
204	Success	No links



POST /api/Churches/AddAttendeeToChurchById

Parameters

No parameters

Request body

application/json

3. { "attendeeID": "3f1a85f9a-5717-4562-b0fc-2c9d2f96af46", "churchID": "3f1a85f9a-5717-4562-b0fc-2c9d2f96af46"}

4. Execute

5. 204

Response headers

date: Sun, 08 May 2022 10:48:45 GMT
server: Kestrel
x-frame-options: NO

Responses

Description	To add Attendee to Church by ID, refer to the figures above for a sequential visual guide starting with label 1 <ol style="list-style-type: none"> 1. Navigate to the Add Attendee to Church by ID module 2. Click “Try it out” 3. Input Attendee ID and Church ID 4. Click execute. 5. 204 return code
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. ChurchesController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features → Churches → Command → AddAttendeeToChurchById Then look at the following files: 1. IChurchRepository.cs 2. AddAttendeeToChurchByIdCommand.cs 3. AddAttendeeToChurchByIdCommandHandler.cs



Infrastructure Layer
src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. ChurchRepository.cs
2. BaseRepository.cs

10.5.9 Remove Attendee From Church By Id

The screenshot shows a list of API endpoints for the 'Churches' endpoint. The 'Remove Attendee From Church By Id' operation is highlighted with a red box and labeled with a red '1'. In the detailed view below, a red box highlights the 'Try it out' button and is labeled with a red '2'.

Churches

- GET /api/Churches/GetAllChurches
- GET /api/Churches/GetChurchById/{id}
- GET /api/Churches/GetAllChurchesByApplicationUserId
- GET /api/Churches/GetAllChurchesByAttendeeId
- POST /api/Churches/CreateChurch
- PATCH /api/Churches/UpdateChurch
- DELETE /api/Churches/DeleteChurch
- POST /api/Churches/AddAttendeeToChurchById
- 1. DELETE /api/Churches/RemoveAttendeeFromChurchById**

DELETE /api/Churches/RemoveAttendeeFromChurchById

Parameters

No parameters

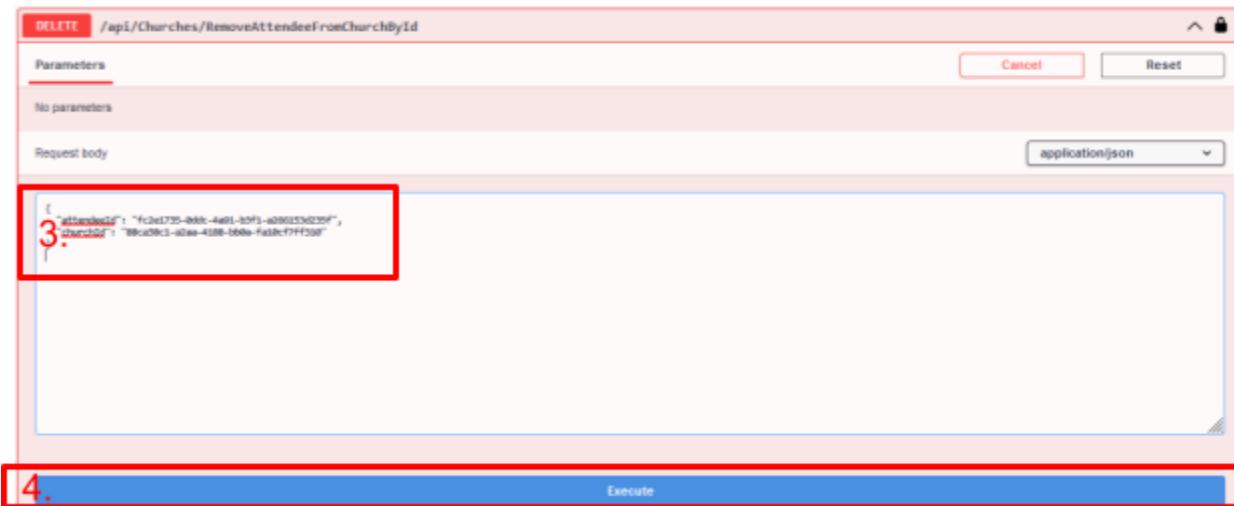
Request body

application/json

Example Value | Schema

```
{ "attendeeId": "3f0a5f04-5717-4562-b5fc-2c963f66af00", "churchId": "3f0a5f04-5717-4562-b5fc-2c963f66af00" }
```





Description	To delete Attendee to Church by ID, refer to the figures above for a sequential visual guide starting with label 1 <ol style="list-style-type: none"> 1. Navigate to the Delete Attendee from Church by ID module 2. Click “Try it out” 3. Input Attendee ID and Church ID 4. Click execute. 5. 204 Code Success
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 2. ChurchesController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features → Churches → Command → RemoveAttendeeFromChurchById



Then look at the following files:

1. IChurchRepository.cs
2. RemoveAttendeeFromChurchByIdCommand.cs
3. RemoveAttendeeFromChurchByIdCommandHandler.cs

Infrastructure Layer
src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. ChurchRepository.cs
2. BaseRepository.cs

10.6 Donation API Functions

10.6.1 Get All Donations

Donation

1.	GET	/api/Donation/GetAllDonations	▼	🔒
	GET	/api/Donation/GetDonationById/{id}	▼	🔒
	GET	/api/Donation/GetAllDonationsByAttendeeId	▼	🔒
	GET	/api/Donation/GetAllDonationsByChurchId	▼	🔒
	POST	/api/Donation/CreateDonation	▼	🔒
	PATCH	/api/Donation/UpdateDonation	▼	🔒
	DELETE	/api/Donation/DeleteDonation	▼	🔒

GET	/api/Donation/GetAllDonations	▲	🔒
Parameters		2.	Try it out
No parameters			



GET /api/Donation/GetAllDonations

Parameters

No parameters

3. Execute

4. Server response

Code	Details
200	Response body <pre>[{ "donationId": "b8ca0e15-e657-48db-8963-8fafd7e51c64", "title": "Donation for Roof", "date": "2022-05-01T19:45:33.5726625-07:00", "amount": 50, "type": "Cash" }, { "donationId": "6238f74b-b22a-488e-b5c7-852317057f7a", "title": "Donation", "date": "2022-05-01T19:45:33.5726631-07:00", "amount": 100, "type": "Zello" }, { "donationId": "86aad8ed-b8ac-45de-9c85-7a999ae08e93", "title": "Donation", "date": "2022-05-01T19:45:33.5726636-07:00", "amount": 15, "type": "Cash" }, { "donationId": "a1512c60-d210-4309-93ac-1ee94f80a0cb", "title": "Payment for Service", "date": "2022-05-01T19:45:33.572664-07:00", "amount": -40, "type": "Check" }]</pre> <p>Download</p>
Response headers	
content-type: application/json; charset=utf-8 date: Mon, 02 May 2022 02:47:40 GMT server: Kestrel	

Description	To get all donations, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the get all donations section under the Donations in the Website API 2. On the right side of the drop down click on “Try it out” 3. Execute the function 4. A 200 Ok Response should be generated with a list of all donations.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers

API Layer

Then look at the following files:

1. DonationController.cs

Application Layer

src → Core → Contracts → Persistence

src → Core → Features → Donation → Query → GetDonationList

Then look at the following files:

1. IDonationRepository.cs
2. DonationListVm.cs
3. GetDonationListQuery.cs
4. GetDonationListQueryHandler.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. DonationRepository.cs

10.6.2 Get Donation By Id

The screenshot shows a list of API endpoints for the 'Donation' controller. The endpoints are:

- GET /api/Donation/GetAllDonations
- 1. GET /api/Donation/GetDonationById/{id} (highlighted with a red border)
- GET /api/Donation/GetAllDonationsByAttendeeId
- GET /api/Donation/GetAllDonationsByChurchId
- POST /api/Donation/CreateDonation
- PATCH /api/Donation/UpdateDonation
- DELETE /api/Donation/DeleteDonation

Each endpoint row includes a lock icon indicating security level.



GET /api/Donation/GetDonationById/{id}

Parameters

Name	Description
id * required string(\$uuid) (path)	id

2. Try it out

GET /api/Donation/GetDonationById/{id}

Parameters

Name	Description
id * required string(\$uuid) (path)	3. id

4. Execute

Cancel

Server response

Code Details

200 Response body

```
{  
  "donationId": "86aad8ed-b0ac-45de-9c85-7a999ae08e93",  
  "title": "Donation",  
  "date": "2022-05-01T19:45:33.5726636-07:00",  
  "amount": 15,  
  "type": "Cash",  
  "description": "Anomous Donation",  
  "churchFK": "162e4e82-ee05-40ea-b0a5-103269ab1d1c",  
  "attendeeFK": null  
}
```

5.

Download

content-type: application/json; charset=utf-8
date: Mon, 02 May 2022 03:02:18 GMT
server: Kestrel

Description	To get all donations by Id, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none">1. Navigate to the get all donations by Id section under the Donations Tab in the Website API2. On the right side of the drop down, click on “Try it out”
--------------------	---



	<p>3. Enter the Donation Id</p> <p>4. A 200 Ok code response should generate with the correct donation tied to the Id.</p>
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. DonationController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Donation → Query → GetDonationDetail</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IDonationRepository.cs 2. DonationDetailVm.cs 3. GetDonationDetailQuery.cs 4. GetDonationDetailQueryHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. DonationRepository.cs



10.6.3 Get All Donations By Attendee Id

Donation

GET	/api/Donation/GetAllDonations	▼ 🔒
GET	/api/Donation/GetDonationById/{id}	▼ 🔒
1. GET	/api/Donation/GetAllDonationsByAttendeeId	▼ 🔒
GET	/api/Donation/GetAllDonationsByChurchId	▼ 🔒
POST	/api/Donation/CreateDonation	▼ 🔒
PATCH	/api/Donation/UpdateDonation	▼ 🔒
DELETE	/api/Donation/DeleteDonation	▼ 🔒

Try it out

Parameters

Name	Description
id	string(\$uuid) (query)

Try it out

Try it out

Parameters

Name	Description
id	string(\$uuid) (query)

Execute



Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7252/api/Donation/GetAllDonationsByAttendeeId?id=b47fbc59-f0f9-4380-b053-b398c23a564e' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJNaWNoYmVsSm9yZGFuIiwianRpIjoiZfd1NGM4MDgtMmQ3Yi06'
```

Request URL

<https://localhost:7252/api/Donation/GetAllDonationsByAttendeeId?id=b47fbc59-f0f9-4380-b053-b398c23a564e>

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "donationId": "c69e1ae7-90e2-4da7-82db-e608b07f3679", "title": "Donation", "date": "2022-05-05T09:41:53.931Z", "amount": 50, "type": "Cash", "churchFK": "a7dbc7f0-d5c8-471c-8b53-71cffdbaa1c9", "attendeeFK": "b47fbc59-f0f9-4380-b053-b398c23a564e" }]</pre> <p style="color: red; margin-left: 20px;">5.</p> <p>Copy Download</p> <p>Response headers</p> <pre>content-type: application/json; charset=utf-8 date: Thu, 05 May 2022 16:46:27 GMT server: Kestrel</pre>

Description	<p>To get all donations by attendee Id, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the get all donation by attendee Id in the Donations tab in the Website API 2. On the right side of the drop down click on “Try it out” 3. Input the Attendee Id for the donation that is trying to be retrieved 4. Execute the Function 5. A 200 ok Response should be generated with the list of donations for that attendee Id.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. DonationController.cs <p>Application Layer</p> <p>src → Core → Contracts → Persistence</p> <p>src → Core → Features → Donation → Query → GetAllDonationsByAttendee</p>



Then look at the following files:

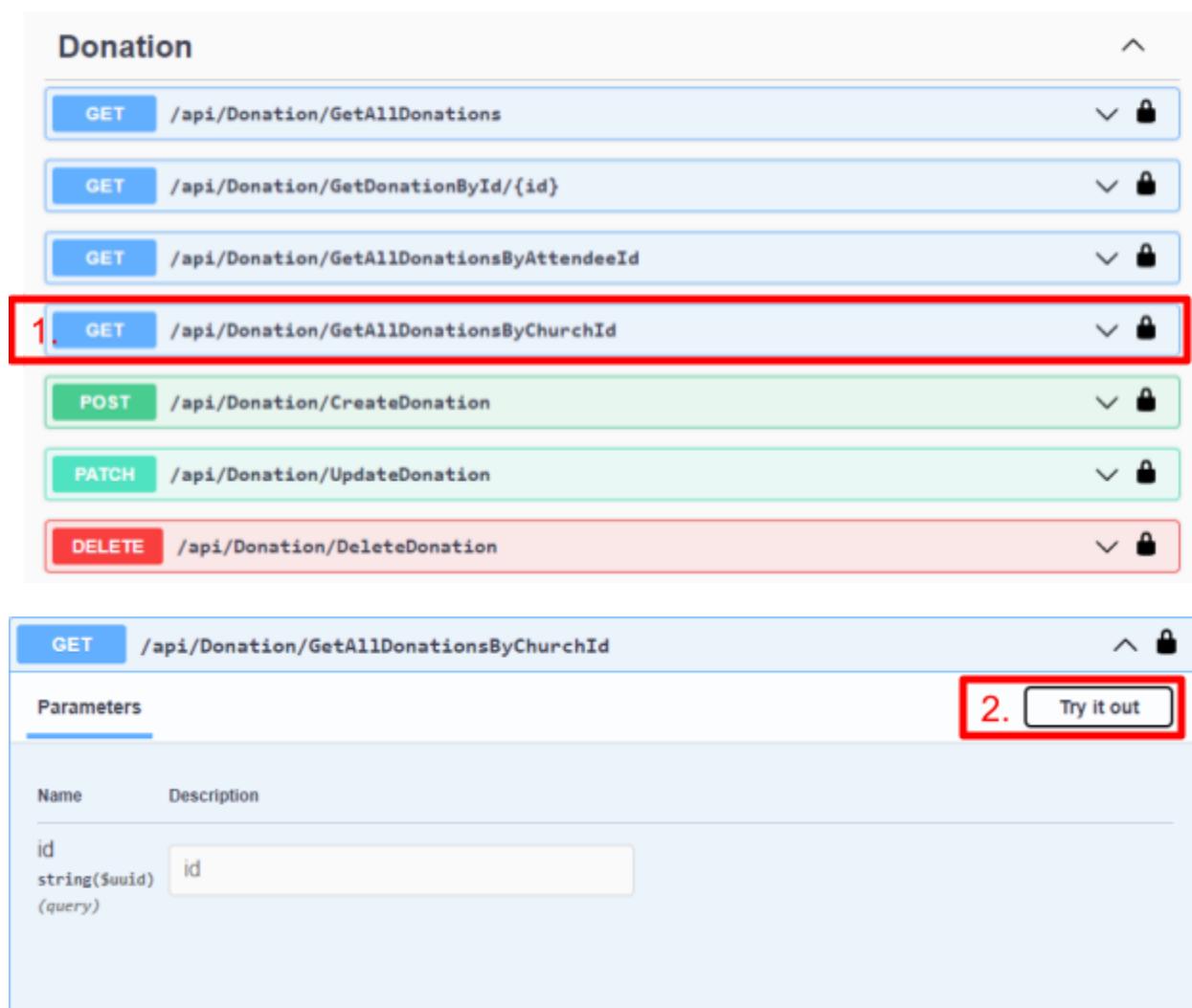
5. IDonationRepository.cs
6. AttendeeDonationsVm.cs
7. GetAllDonationsByAttendeeQuery.cs
8. GetAllDonationsByAttendeeQueryHandler.cs

Infrastructure Layer
src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

3. DonationRepository.cs

10.6.4 Get All Donations By Church Id



Donation

1. GET /api/Donation/GetAllDonationsByChurchId

2. Try it out

Parameters

Name	Description
id	string(\$uuid) (query)

GET /api/Donation/GetAllDonationsByChurchId

Parameters

Name Description

id string(\$uuid)
(query) 3. id

4. Execute Clear

Server response

Code	Details
200	<p>Response body</p> <pre>[{ "donationId": "86aad8ed-b0ac-45de-9c85-7a999ae08e93", "title": "Donation", "date": "2022-05-01T19:45:33.5726636-07:00", "amount": 15, "type": "Cash", "churchFK": "162e4e82-ee05-40ea-b0a5-103269ab1d1c" }, { "donationId": "b8ca0e15-e657-48db-8963-8fafd7e51c64", "title": "Donation for Roof", "date": "2022-05-01T19:45:33.5726625-07:00", "amount": 50, "type": "Cash", "churchFK": "162e4e82-ee05-40ea-b0a5-103269ab1d1c" }]</pre> <p>5.</p> <p>Download</p>

Description	To get all donations by church Id, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the get all donations by Church Id under the Donations tab on the Website API 2. On the right side of the drop down click on “Try it out” 3. Enter the church Id to find the donations 4. Execute the function 5. 200 Ok Response should be created with the donations for the church.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. DonationController.cs



Application Layer
src → Core → Contracts → Persistence
src → Core → Features → Donation → Query → CreateDonation

Then look at the following files:

1. IDonationRepository.cs
2. ChurchDonationsVm.cs
3. GetAllDonationsByChurchQuery.cs
4. GetAllDonationsByChurchQueryHandler.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. DonationRepository.cs

10.6.5 Create Donation

The screenshot shows a list of API endpoints for the 'Donation' resource. The 'Create Donation' endpoint, which is a POST request to '/api/Donation/CreateDonation', is highlighted with a red box and labeled '1.' to its left. Below this, there is a detailed view for the 'Create Donation' endpoint, also highlighted with a red box and labeled '2.' to its left. This detailed view includes sections for 'Parameters' (which states 'No parameters') and a 'Try it out' button.

Donation	
GET	/api/Donation/GetAllDonations
GET	/api/Donation/GetDonationById/{id}
GET	/api/Donation/GetAllDonationsByAttendeeId
GET	/api/Donation/GetAllDonationsByChurchId
1. POST	/api/Donation/CreateDonation
PATCH	/api/Donation/UpdateDonation
DELETE	/api/Donation/DeleteDonation

POST /api/Donation/CreateDonation	
Parameters	2. Try it out
No parameters	



POST /api/Donation/CreateDonation

Parameters

No parameters

Request body

3. application/json

4.

```
{
  "title": "string",
  "date": "2022-05-05T16:15:25.090Z",
  "amount": 0,
  "type": "string",
  "description": "string",
  "churchFK": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "attendeeFK": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}
```

5. Execute

Responses

Curl

```
curl -X 'POST' \
'https://localhost:7252/api/Donation/CreateDonation' \
-H 'accept: text/plain' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJzdWIiOiJ0ZXN0VXNlciIsImp0aSI6Ijg1Y2RizTQ0LNWk0WtNDM3Ny05 \
-H 'Content-Type: application/json' \
-d '{
  "title": "Giving Thanks",
  "date": "2022-05-05T16:17:47.764Z",
  "amount": 10000,
  "type": "Donation",
  "description": "string",
  "churchFK": "77b03c30-a472-4264-ac35-1beb7a5acdb0",
  "attendeeFK": "fc2e1735-0ddc-4a91-b3f1-a266153d235f"
}'
```

Request URL

https://localhost:7252/api/Donation/CreateDonation

Server response

Code	Details
201	Response body 6. "de124b1a-5a75-4e72-d7a1-08da2eb2c59e"

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 05 May 2022 16:18:41 GMT
server: Kestrel
```



Description	<p>To create a donation, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the create donation section under the donation tab in the Website API. 2. On the right side of the drop down click on “Try it out” 3. Make sure the parameter drop down is set to “application/json” 4. Input the correct data for the donation that is being created 5. execute the command 6. A 201 Created response should appear indicating the donation was successfully made along with its own donationId.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. DonationController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Donation → Command → CreateDonation</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IDonationRepository.cs 2. CreateDonationCommand.cs 3. CreateDonationCommandHandler.cs 4. CreateDonationCommandValidator.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. DonationRepository.cs 2. BaseRepository.cs



10.6.6 Update Donation

Donation

GET	/api/Donation/GetAllDonations	▼ 🔒
GET	/api/Donation/GetDonationById/{id}	▼ 🔒
GET	/api/Donation/GetAllDonationsByAttendeeId	▼ 🔒
GET	/api/Donation/GetAllDonationsByChurchId	▼ 🔒
POST	/api/Donation/CreateDonation	▼ 🔒
1. PATCH	/api/Donation/UpdateDonation	▼ 🔒
DELETE	/api/Donation/DeleteDonation	▼ 🔒

PATCH /api/Donation/UpdateDonation

Parameters

No parameters

2. Try it out

PATCH /api/Donation/UpdateDonation

Parameters

No parameters

Request body

3. application/json

{
 "donationId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
 "title": "string",
 "date": "2022-05-05T16:26:53.916Z",
 4. "amount": 0,
 "type": "string",
 "description": "string"
}

5. Execute

Responses

Curl

```
curl -X 'PATCH' \  
  'https://localhost:7252/api/Donation/UpdateDonation' \  
  -H 'accept: */*' \  
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJzdWIiOiJ0ZXN0VXNlciIsImp0aSI6Ijg1Y2RiZTQ0LWNkOWMtNDM3Ny05' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "donationId": "de124bia-5a75-4e72-d7a1-08da2eb2c59e",  
    "title": "Giving More",  
    "date": "2022-05-05T16:26:53.916Z",  
    "amount": 10000,  
    "type": "Donation",  
    "description": "string"  
}'
```

Request URL

<https://localhost:7252/api/Donation/UpdateDonation>

Server response

Code Details

204
6. Response headers

date: Thu,05 May 2022 16:29:35 GMT
server: Kestrel



Description	<p>To update a donation, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the update donation section under the donations tab in the Website API 2. On the right side of the drop down click on “Try it out” 3. Make sure the parameter drop down is set to “application/json” 4. Input the necessary information on the donation that needs to be updated 5. Execute the function 6. A 204 No Content code should be generated indicating that the update was successful.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. DonationController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Donation → Command → UpdateDonation</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IDonationRepository.cs 2. UpdateDonationCommand.cs 3. UpdateDonationCommandHandler.cs 4. UpdateDonationCommandValidator.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. DonationRepository.cs 2. BaseRepository.cs



10.6.7 Delete Donation

Donation

GET	/api/Donation/GetAllDonations	▼ 🔒
GET	/api/Donation/GetDonationById/{id}	▼ 🔒
GET	/api/Donation/GetAllDonationsByAttendeeId	▼ 🔒
GET	/api/Donation/GetAllDonationsByChurchId	▼ 🔒
POST	/api/Donation/CreateDonation	▼ 🔒
PATCH	/api/Donation/UpdateDonation	▼ 🔒
1. DELETE	/api/Donation/DeleteDonation	▼ 🔒

DELETE /api/Donation/DeleteDonation

Parameters

Name	Description
Id string(\$uuid) (query)	Id

2. Try it out

DELETE /api/Donation/DeleteDonation

Parameters

Name	Description
Id string(\$uuid) (query)	3. Id

4. Execute



Responses

Curl

```
curl -X 'DELETE' \
  'https://localhost:7252/api/Donation/DeleteDonation?Id=de124b1a-5a75-4e72-d7a1-08da2eb2c59e' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJ0ZXN0VXNlcjIiImp0aSI6Ijg1Y2RiZTQ0LWNkOWMtNDk3Ny05'
```

Request URL

<https://localhost:7252/api/Donation/DeleteDonation?Id=de124b1a-5a75-4e72-d7a1-08da2eb2c59e>

Server response

Code	Details
204 5.	Response headers date: Thu, 05 May 2022 16:37:20 GMT server: Kestrel

Description	<p>To delete a donation, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the delete a donation section under the donations tab in the Website API 2. On the right side of the drop down click on “Try it out” 3. Input the Id of the donation which needs to be deleted 4. Execute the function 5. A 204 no content response should be generated indication the donation has been deleted
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. DonationController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Donation → Command → DeleteDonation</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IDonationRepository.cs 2. DeleteDonationCommand.cs 3. DeleteDonationCommandHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p>



Then look at the following files:

1. DonationRepository.cs
2. BaseRepository.cs

10.7 Event Features

10.7.1 Get All Events

The screenshot shows a list of API endpoints for the 'Event' endpoint. The first endpoint, 'GET /api/Event/GetAllEvents', is highlighted with a red box and a red number '1'. The 'Try it out' button for this specific endpoint is also highlighted with a red box and a red number '2'.

Method	Endpoint	Status
GET	/api/Event/GetAllEvents	✓
GET	/api/Event/GetEventById/{id}	✓
GET	/api/Event/GetAllEventsByAttendeeId	✓
GET	/api/Event/GetAllEventsByMinistryServiceId	✓
GET	/api/Event/GetAllEventsByChurchId	✓
POST	/api/Event/CreateEventFromChurch	✓
POST	/api/Event/CreateEventFromMinistryService	✓
PATCH	/api/Event/UpdateEvent	✓
DELETE	/api/Event/DeleteEvent	✓
POST	/api/Event/AddAttendeeToEventId	✓
POST	/api/Event/RemoveAttendeeFromEventId	✓



Parameters

No parameters

3. Execute

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7252/api/events/getAllEvents' \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cC...'
```

Request URL

<https://localhost:7252/api/events/getAllEvents>

Server response

Code Details

200 Response body

```
[
  {
    "eventID": "347235ce-04d9-4cb3-aedf-0e0351040cc8",
    "overdue": "Event",
    "start": "2022-04-24T08:01:47.788425-07:00",
    "end": "2022-04-24T08:01:47.788434-07:00",
    "overchanger": "Pastor"
  },
  {
    "eventID": "123456-7890-4321-5678-901234567890",
    "overdue": "Meeting group service",
    "start": "2022-04-24T08:01:47.788434-07:00",
    "end": "2022-04-24T08:01:47.788433-07:00",
    "overchanger": "Worship"
  },
  {
    "eventID": "987654-3210-4987-6543-234567890123",
    "overdue": "HS Worship Service",
    "start": "2022-04-24T08:01:47.788433-07:00",
    "end": "2022-04-24T08:01:47.788434-07:00",
    "overchanger": "PAC"
  },
  {
    "eventID": "123456-7890-4321-5678-901234567890",
    "overdue": "Worship",
    "start": "2022-04-24T08:01:47.788432-07:00",
    "end": "2022-04-24T08:01:47.788433-07:00",
    "overchanger": "Worship"
  }
]
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 26 May 2022 16:00:00 GMT
server: Kestrel
x-frame-options: SAMEORIGIN
```

Message

Description	To get all Events, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the Get All Events module 2. Click Try it Out on the top right corner 3. Click Execute 4. View output
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. EventController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features → Events → Query → GetEventList Then look at the following files:



1. IEventRepository.cs
2. EventListVm.cs
3. GetEventListQuery.cs
4. GetEventListQueryHandler.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. EventRepository.cs

10.7.2 Get Event By Id

Event

GET	/api/Event/GetAllEvents	▼ 🔒
1 GET	/api/Event/GetEventById/{id}	▼ 🔒
GET	/api/Event/GetAllEventsByAttendeeId	▼ 🔒
GET	/api/Event/GetAllEventsByMinistryServiceId	▼ 🔒
GET	/api/Event/GetAllEventsByChurchId	▼ 🔒
POST	/api/Event/CreateEventFromChurch	▼ 🔒
POST	/api/Event/CreateEventFromMinistryService	▼ 🔒
PATCH	/api/Event/UpdateEvent	▼ 🔒
DELETE	/api/Event/DeleteEvent	▼ 🔒
POST	/api/Event/AddAttendeeToEventId	▼ 🔒
POST	/api/Event/RemoveAttendeeFromEventId	▼ 🔒

GET /api/Event/GetEventById/{id}

Parameters

2. Try it out

Name	Description
id * required	string(\$uuid) (path)
id	id



GET /api/Event/GetEventById/{id}

Parameters

Name	Description
id * required string(\$uuId) (path)	3. ID

Execute

Responses

Curl:

```
curl -X 'GET' \
  'https://localhost:7282/api/Event/GetEventById/247235ce-e4a9-4c9c-aaed-0492b194bce8' \
  -H 'accept: application/json' \
  -H 'Authorization: bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJsb2dpbi5jb20udG8gZW1haWwuc2VydmljZS5jb20iLCJleHAiOjE2NjQyOTMxNjMsIm5iZiI6MTY2NDI5MzE2M30.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6ImltYWdlciBhbmQgZW1haWwgdG8gZW1haWwuc2VydmljZS5jb20iLCJpYXQiOjE2NjQyOTMxNjMsImV4cCI6MTY2NDI5MzE2M30' --insecure
```

Request URL:
<https://localhost:7282/api/Event/GetEventById/247235ce-e4a9-4c9c-aaed-0492b194bce8>

Server response

Code	Details
200	Response body <pre>{ "eventId": "247235ce-e4a9-4c9c-aaed-0492b194bce8", "eventTitle": "Walk", "start": "2022-04-04T08:00:00Z", "end": "2022-04-04T08:00:00Z", "eventAddress": "Walker", "phoneNumber": "319-481-4600", "description": "Donation Event" }</pre> <p>5. Download</p> Response headers <pre>content-type: application/json; charset=UTF-8 date: Wed, 04 May 2022 03:54:25 GMT server: Kestrel x-ffrfr-splv: 12</pre>

Description	To get Events by eventID, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the Get Event by Id module 2. Click Try it out on the top right corner 3. Input Event ID 4. Click Execute 5. Verify output data
--------------------	--



<p>List of Code Files</p> <p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Events → Query → GetEventDetail</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IEventRepository.cs 2. EventDetailVm.cs 3. GetEventDetailQuery.cs 4. GetEventDetailQueryHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventRepository.cs

10.7.3 Get All Events By Attendee Id

The screenshot shows a list of API endpoints for the 'Event' resource. The endpoints are:

- GET /api/Event/GetAllEvents
- GET /api/Event/GetEventById/{id}
- 1 GET /api/Event/GetAllEventsByAttendeeId
- GET /api/Event/GetAllEventsByMinistryServiceId
- GET /api/Event/GetAllEventsByChurchId
- POST /api/Event/CreateEventFromChurch
- POST /api/Event/CreateEventFromMinistryService
- PATCH /api/Event/UpdateEvent
- DELETE /api/Event/DeleteEvent
- POST /api/Event/AddAttendeeToEventId
- POST /api/Event/RemoveAttendeeFromEventId



GET /api/Event/GetAllEventsByAttendeeId

Parameters

Name	Description
id	string(uuid) (queried)

Responses

Code	Description	Links
200	Success	No links

Media type: text/plain

Example Value: Schema

```
{ "eventId": "3b0c9036-a050-4f82-ab71-2d58a5fbca98", "start": "2022-05-06T05:47:47.000Z", "end": "2022-05-06T05:47:47.000Z", "eventHandler": "String", "phoneNumber": "String" }
```

Parameters

Name	Description
id	string(uuid) (queried)

Responses

Curl:

```
curl -X 'GET' \
  'https://localhost:7252/api/event/GetAllEventsByAttendeeId?id=3b0c9036-a050-4f82-ab71-2d58a5fbca98' \
  -H 'accept: text/plain' \
  -H 'Authorization: bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6ImltYWdlcyBvZiIsInVzZXJuYW1lIjoiQWxhZGRpbmcuQ29yZS5jb20iLCJleHAiOjE2NjUwOTQwMjAsImNvbmNob3ciOiIzYjBjOTAyNgIifQ' --compressed
```

Request URL: https://localhost:7252/api/event/GetAllEventsByAttendeeId?id=3b0c9036-a050-4f82-ab71-2d58a5fbca98

Server response:

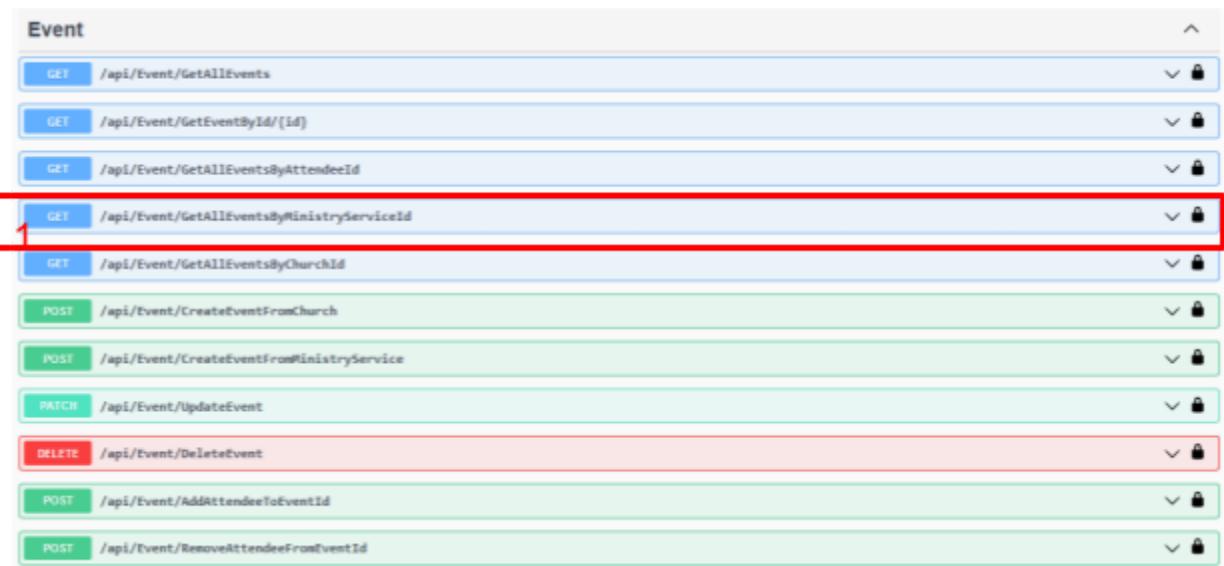
Code	Details
200	Response body: [{ "eventId": "3b0c9036-a050-4f82-ab71-2d58a5fbca98", "start": "2022-05-06T05:47:47.000Z", "end": "2022-05-06T05:47:47.000Z", "eventHandler": "String", "phoneNumber": "String" }] Response headers: content-type: application/json; charset=UTF-8 date: Wed, 24 May 2022 05:59:43 GMT server: Kestrel x-azure-applid: 16

Description	To get Events by AttendeeId, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none">1. Navigate to the Get Event by Id module2. Click Try it out on the top right corner
--------------------	---



	<p>3. Input Attendee ID 4. Click Execute 5. Verify output data</p>
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files: 1. EventController.cs</p> <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Events → Query → GetAllEventsByAttendeeId</p> <p>Then look at the following files: 1. IEventRepository.cs 2. EventsAttendeeListVm.cs 3. GetAllEventsByAttendeeIdQuery.cs 4. GetAllEventsByAttendeeIdQueryHandler.cs</p> <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files: 1. EventRepository.cs</p>

10.7.4 Get All Event By Ministry Service Id



The screenshot shows the Network tab of the F12 developer tools. It lists various API endpoints for the 'Event' controller. The endpoint '/api/Event/GetAllEventsByMinistryServiceId' is highlighted with a red box and a red number '1'.

Event		
GET	/api/Event/GetAllEvents	▼ 🔒
GET	/api/Event/GetEventById/{id}	▼ 🔒
GET	/api/Event/GetAllEventsByAttendeeId	▼ 🔒
1 GET	/api/Event/GetAllEventsByMinistryServiceId	▼ 🔒
GET	/api/Event/GetAllEventsByChurchId	▼ 🔒
POST	/api/Event/CreateEventFromChurch	▼ 🔒
POST	/api/Event/CreateEventFromMinistryService	▼ 🔒
PATCH	/api/Event/UpdateEvent	▼ 🔒
DELETE	/api/Event/DeleteEvent	▼ 🔒
POST	/api/Event/AddAttendeeToEventId	▼ 🔒
POST	/api/Event/RemoveAttendeeFromEventId	▼ 🔒



GET /api/Event/GetAllEventsByMinistryServiceId

Parameters

Name	Description
id	string(Signed) (query)

Responses

Code	Description	Links
200	Success	No links

Media type: text/plain

Example Value: Schemas

```
{
  "eventId": "3f4a9f6a-5717-4962-b0fc-219d9f66a6a",
  "start": "2022-05-06T04:00:23.999Z",
  "end": "2022-05-06T04:00:23.999Z",
  "eventMessage": "String",
  "phoneNumber": "String"
}
```

Parameters

Name	Description
id	string(UUID) (query)

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7252/api/Event/GetAllEventsByMinistryServiceId?id=f525dfbe-1106-4881-903e-22753f03e398' \
  -H 'accept: text/plain' \
  -H 'Authorization: bearer eyJhbGciOiJIUzI1NiIsInR5cCIsImtpZCIiLCJ0eXAiOiJKV1QiLCJhdWQiOiIxMjUxMDUyNjYwLTAzMDQsMSIsImF1ZGUiOiJ1c2VycG9yIiwiaWF0IjoxNjIyOTk4NTUwLCJleHAiOjE2NjI5OTk4NTYgZmlyc2lvbgJ9' --insecure
```

Request URL: <https://localhost:7252/api/Event/GetAllEventsByMinistryServiceId?id=f525dfbe-1106-4881-903e-22753f03e398>

Server response

Code	Details
200	Response body

```
[
  {
    "eventId": "f525dfbe-1106-4881-903e-22753f03e398",
    "start": "2022-05-06T04:00:23.999Z",
    "end": "2022-05-06T04:00:23.999Z",
    "eventMessage": "String",
    "phoneNumber": "String"
  }
]
```

Response headers:

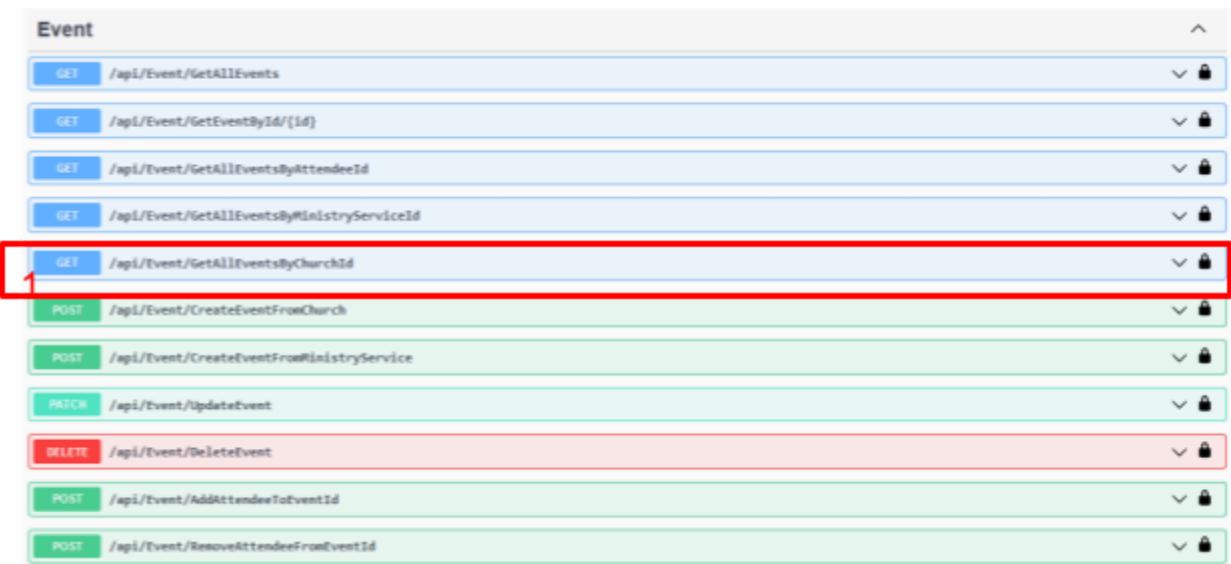
```
Content-Type: application/json; charset=utf-8
Date: Wed, 16 May 2023 06:46:52 GMT
Server: Kestrel
x-Frame-Options: no
```

Description	To get Events by MinistryServiceId, refer to the figures above for a sequential visual guide starting with label 1.
	<ol style="list-style-type: none"> 1. Navigate to the Get Event by Id module 2. Click Try it out on the top right corner



	<p>3. Input Ministry ServiceID 4. Click Execute 5. Verify output data</p>
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Events → Query → GetEventsByMinistryServiceId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IEventRepository.cs 2. EventMinistryServiceListVm.cs 3. GetEventsByMinistryServiceIdQuery.cs 4. GetEventsByMinistryServiceIdQueryHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventRepository.cs

10.7.5 Get All Events By Church Id



The screenshot shows a list of API endpoints for the 'Event' resource. The endpoint `/api/Event/GetAllEventsByChurchId` is highlighted with a red box. The other endpoints listed are:

- GET /api/Event/GetAllEvents
- GET /api/Event/GetEventById/{id}
- GET /api/Event/GetAllEventsByAttendeeId
- GET /api/Event/GetAllEventsByMinistryServiceId
- GET /api/Event/GetAllEventsByChurchId
- POST /api/Event/CreateEventFromChurch
- POST /api/Event/CreateEventFromMinistryService
- PATCH /api/Event/UpdateEvent
- DELETE /api/Event/DeleteEvent
- POST /api/Event/AddAttendeeToEventId
- POST /api/Event/RemoveAttendeeFromEventId



GET /api/Event/GetAllEventsByChurchId

Parameters

Name Description

`id` string(id) `id` string(id) `id` string(id)

Responses

Code Description Links

200 Success No links

Media type

text/plain

Controls Accept header

Example Value Schema

```
[{"eventid": "3fa85f64-07d7-4960-bfc-2e063f60efef", "start": "2023-05-01T00:00:00Z", "end": "2023-05-04T00:00:00Z", "eventbargain": "string", "phonenumbe": "string"}]
```

Parameters

Cancel

Name	Description
id string((kuid)) (query)	a10a44e5-1b6b-4fa8-ab30-4f3385b22399

4. **Execute** **Clear**

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7252/api/Event/GetAllEventsByChurchId?id=a10a44e5-1b6b-4fa8-ab30-f03385b22399' \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCJ9.DjWV1C5DpN5TEJZUHfQULFNT0EwLTHhOzLThEYmJnqgzzsowMjZjyjvzvqLwvTnQ1LwAstsDnyC3EJL0MWhQdLqPndkYy'
```

Request URL

```
https://localhost:7252/api/Event/GetAllEventsByChurchId?id=a10a44e5-1b6b-4fa8-ab30-f03385b22399
```

Server response

Code Details

200 Response body

5.

```
{
  "eventId": "247335e-e449-45c9-aed3-04030194ce09",
  "start": "2022-04-24T08:07:47.798825-07:00",
  "end": "2022-04-24T09:07:47.798824-07:00",
  "eventManager": "Pastor",
  "phoneNumber": "319-481-4667"
},
{
  "eventId": "1328b6ad-7551-4646-933f-5905a38921a7",
  "start": "2022-04-24T08:07:47.798825-07:00",
  "end": "2022-04-24T09:07:47.798824-07:00",
  "eventManager": "Stephen",
  "phoneNumber": "319-481-4667"
},
{
  "eventId": "ae503aef-7571-42c5-93be-61589e050ec5",
  "start": "2022-04-24T08:07:47.798825-07:00",
  "end": "2022-04-24T09:07:47.798824-07:00",
  "eventManager": "Pastor",
  "phoneNumber": "319-481-4667"
}
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 04 May 2022 00:00:20 GMT
server: Kestrel
x-frame-options: SAMEORIGIN
```

Description	<p>To get Events by ChurchId, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the Get Event by Id module 2. Click Try it out on the top right corner 3. Input Ministry ServiceID 4. Click Execute 5. Verify output data
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Events → Query → GetEventsByChurchId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IEventRepository.cs 2. EventChurchListVm.cs 3. GetEventsByChurchIdQuery.cs 4. GetEventsByChurchIdQueryHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventRepository.cs



10.7.6 Create Event From Church

Event

GET	/api/Event/GetAllEvents	v 🔒
GET	/api/Event/GetEventById/{id}	v 🔒
GET	/api/Event/GetAllEventsByAttendeeId	v 🔒
GET	/api/Event/GetAllEventsByMinistryServiceId	v 🔒
GET	/api/Event/GetAllEventsByChurchId	v 🔒
1 POST	/api/Event/CreateEventFromChurch	v 🔒
POST	/api/Event/CreateEventFromMinistryService	v 🔒
PATCH	/api/Event/UpdateEvent	v 🔒
DELETE	/api/Event/DeleteEvent	v 🔒
POST	/api/Event/AddAttendeeToEventId	v 🔒
POST	/api/Event/RemoveAttendeeFromEventId	v 🔒

POST /api/Event/CreateEventFromChurch

Parameters

No parameters

Request body

application/json

Try it out

Reset

Example Value | Schema

```
{ "churchId": "aefc1034-f5bc-4f45-16ff-00d9e07ec558", "eventTitle": "string", "start": "2022-05-04T00:00:38.630Z", "end": "2022-05-04T04:00:38.630Z", "recurring": true, "eventNotes": "string", "phoneNumber": "string", "description": "string" }
```



The screenshot shows the "Create an event from a church" section of the ChurchPoint API. Step 3 highlights the JSON request body with a red box, containing:

```
[{"church": "elijahs-fake-church-which-does-not-exist.com", "eventname": "Donation Event", "start": "2022-05-01T00:00:00Z", "end": "2022-05-01T00:10:00Z", "recurring": false, "eventholder": "Pastor John", "phonetholder": "123-456-7894", "description": "Event for more donation"}]
```

Step 4 highlights the "Execute" button with a red box.

Responses

Curl

```
curl -X "POST" \
  "https://localhost:7212/api/event/CreateEventFromChurch" \
  -H "accept: text/plain" \
  -H "Authorization: Bearer eyJhbGciOiJIUzI1NiJ9.yA" \
  -H "Content-Type: application/json" \
  -d \
  [{"church": "elijahs-fake-church-which-does-not-exist.com", "eventname": "Donation Event", "start": "2022-05-01T00:00:00Z", "end": "2022-05-01T00:10:00Z", "recurring": false, "eventholder": "Pastor John", "phonetholder": "123-456-7894", "description": "Event for more donation"}]
```

Request URL

https://localhost:7212/api/event/CreateEventFromChurch

Server response

201

Response body

```
"20220501-0010-001f-000e-000000000000"
```

Download

Response headers

```
content-type: application/json; charset=utf-8
date: Wed, 04 May 2022 00:10:26 GMT
server: Kestrel
x-forefront-reply: N
```

Description	To create an event from a Church, refer to the figures above for a sequential visual guide starting with label 1. <ol style="list-style-type: none"> 1. Navigate to the create an event from a church section under the event tab in the Website API 2. On the right side of the drop down click on “Try it out” 3. Make sure that the response body is in application/json and input the information regarding the new event with the correct church foreign key. 4. Execute the Command 5. A 201 Created response will be generated with the Event Id
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers



API Layer

Then look at the following files:

1. EventController.cs

Application Layer

src → Core → Contracts → Persistence

src → Core → Features → Events → Command → CreateEventFromChurch

Then look at the following files:

1. IEventRepository.cs
2. CreateEventFromChurchCommand.cs
3. CreateEventFromChurchCommandHandler.cs
4. CreateEventFromChurchCommandValidator.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. EventRepository.cs
2. BaseRepository.cs

10.7.7 Create Event From Ministry Service

Event		
GET	/api/Event/GetAllEvents	^
GET	/api/Event/GetEventById/{id}	^
GET	/api/Event/GetAllEventsByAttendeeId	^
GET	/api/Event/GetAllEventsByMinistryServiceId	^
GET	/api/Event/GetAllEventsByChurchId	^
POST	/api/Event/CreateEventFromChurch	^
POST	/api/Event/CreateEventFromMinistryService	^
PATCH	/api/Event/UpdateEvent	^
DELETE	/api/Event/DeleteEvent	^
POST	/api/Event/AddAttendeeToEventId	^
POST	/api/Event/RemoveAttendeeFromEventId	^



POST /api/Event/CreateEventFromMinistryService

Parameters

No parameters

Try it out

POST /api/Event/CreateEventFromMinistryService

Parameters

No parameters

Cancel Reset

Request body

application/json

{
 "ministryServiceFK": "3fa85f64-5717-4562-b3fc-2c963f66af6",
 "eventName": "string",
 "start": "2022-05-05T17:09:39.480Z",
 "end": "2022-05-05T17:09:39.480Z",
 "reoccurring": true,
 "eventManager": "string",
 "phoneNumber": "string",
 "description": "string"
}

Execute

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7252/api/Event/CreateEventFromMinistryService' \
  -H 'accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJNaWNoYVVsSm9yZGFuIiwiZW1haWf0IjoiN2FhMzE5YTgtM2Q3ZS00
  -H 'Content-Type: application/json' \
  -d '{
    "ministryServiceFK": "f625dfbe-1106-4881-903e-22753f83e398",
    "eventName": "Let Us Rise",
    "start": "2022-05-05T17:09:39.480Z",
    "end": "2022-06-05T17:09:39.480Z",
    "reoccurring": false,
    "eventManager": "Michael Jordan",
    "phoneNumber": "916-798-9875",
    "description": "Time"
  }'
```

Request URL

<https://localhost:7252/api/Event/CreateEventFromMinistryService>

Server response

Code Details

201

Response body

6.

"1b03a8f5-edca-4120-87e5-08da2eb94fdb"



Download

Response headers

```
content-type: application/json; charset=utf-8
date: Thu, 05 May 2022 17:12:31 GMT
server: Kestrel
```

Description To create an event from a ministry service, refer to the figures above for a sequential visual guide starting with label 1.

1. Navigate to the create event from ministry service tab in the Event tab within the website API
2. On the right side of the drop down click on “Try it out”
3. Make sure the request body is “application/json”
4. Within the request body input the ministry service foreign key to create the event
5. Execute the function
6. An event Id should be created with a 201 Created response

List of Code Files

If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers

API Layer

Then look at the following files:

1. EventController.cs



Application Layer
src → Core → Contracts → Persistence
src → Core → Features → Events → Command → CreateEventFromMinistryService

Then look at the following files:

1. IEventRepository.cs
2. CreateEventFromMinistryServiceCommand.cs
3. CreateEventFromMinistryServiceCommandHandler.cs
4. CreateEventFromMinistryServiceCommandValidator.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. EventRepository.cs
2. BaseRepository.cs

10.7.8 Update Event

The screenshot shows the 'Event' endpoint in a Swagger UI. The endpoint is a PATCH method at the URL /api/Event/UpdateEvent. This specific endpoint is highlighted with a red box and has a red number '1' next to it. To the right of the endpoint, there is a 'Try it out' button, which is also highlighted with a red box and has a red number '2.' next to it. Below the endpoint, there is a section for 'Parameters' which states 'No parameters'.



PATCH /api/Event/UpdateEvent

Parameters

No parameters

Request body

3. application/json

4. {
 "eventId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
 "start": "2022-05-05T17:18:44.854Z",
 "end": "2022-05-05T17:18:44.854Z",
 "reoccuring": true,
 "eventManager": "string",
 "phoneNumber": "string",
 "description": "string"
}

5. Execute

Responses

Curl

```
curl -X 'PATCH' \  
'https://localhost:7252/api/Event/UpdateEvent' \  
-H 'accept: */*' \  
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJNaWNoYWwSaW9yZGFuIiwianRpIjoiN2FhMzE5YTgtMDQ3ZS00 \  
-H 'Content-Type: application/json' \  
-d '{  
    "eventId": "1b03a8f5-edca-4120-87e5-08da2eb94fdb",  
    "start": "2022-05-05T17:18:44.854Z",  
    "end": "2022-06-05T17:18:44.854Z",  
    "reoccuring": true,  
    "eventManager": "Marcus Jordan",  
    "phoneNumber": "string",  
    "description": "string"  
}'
```

Request URL

<https://localhost:7252/api/Event/UpdateEvent>

Server response

Code	Details
204	6. Response headers date: Thu, 05 May 2022 17:20:57 GMT server: Kestrel

Description	<p>To update an event, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the Update Event section under the Event tab within the website API. 2. On the right side of the drop down click on “Try it out” 3. Make sure the request body is “application/json” 4. Input the event Id with the information that needs to be updated 5. Execute the Command 6. A 204 No content response should be generated indicating that the event was updated.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Events → Command → UpdateEvent</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IEventRepository.cs 2. UpdateEventCommand.cs 3. UpdateEventCommandHandler.cs 4. UpdateEventCommandValidator.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventRepository.cs 2. BaseRepository.cs



10.7.9 Delete Event

Event

GET	/api/Event/GetAllEvents	v 🔒
GET	/api/Event/GetEventById/{id}	v 🔒
GET	/api/Event/GetAllEventsByAttendeeId	v 🔒
GET	/api/Event/GetAllEventsByMinistryServiceId	v 🔒
GET	/api/Event/GetAllEventsByChurchId	v 🔒
POST	/api/Event/CreateEventFromChurch	v 🔒
POST	/api/Event/CreateEventFromMinistryService	v 🔒
PATCH	/api/Event/UpdateEvent	v 🔒
1. DELETE	/api/Event/DeleteEvent	v 🔒
POST	/api/Event/AddAttendeeToEventId	v 🔒
POST	/api/Event/RemoveAttendeeFromEventId	v 🔒

DELETE /api/Event/DeleteEvent

Parameters

Name	Description
id	string(\$uuid) (query)

2. Try it out

DELETE /api/Event/DeleteEvent

Parameters

Name	Description
id	string(\$uuid) (query)

3. id

4. Execute



Responses

Curl

```
curl -X 'DELETE' \
'https://localhost:7252/api/Event/DeleteEvent?id=1b03a8f5-edca-4120-87e5-08da2eb94fdb' \
-H 'accept: */*' \
-H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpxVCJ9eyJzdWIiOiJNaMNoYmVsSm9yZGFuIiwianRpIjoiN2FhMzE5YTgtM2Q3ZS00LjAifQ'
```

Request URL

```
https://localhost:7252/api/Event/DeleteEvent?id=1b03a8f5-edca-4120-87e5-08da2eb94fdb
```

Server response

Code	Details
204	Response headers
5.	<code>date: Thu,05 May 2022 17:27:00 GMT server: Kestrel</code>

Description	To delete an event, refer to the figures above for a sequential visual guide starting with label 1. 1. Navigate to the delete event tab under the event section in the website API 2. On the right side of the drop down click on “Try it out” 3. Input the Id of the Event that needs to be deleted 4. Execute Funcion 5. A 204 No Content should be generated indicating that the event has been deleted.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. EventController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features→ Events→ Command → DeleteEvent Then look at the following files: 1. IEventRepository.cs 2. DeleteEventCommand.cs 3. DeleteEventCommandHandler.cs Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories Then look at the following files:



- | | |
|--|---|
| | 4. EventRepository.cs
5. BaseRepository.cs |
|--|---|

10.7.10 Add Attendee To Event Id

Event

Method	Endpoint	Lock
GET	/api/Event/GetAllEvents	🔒
GET	/api/Event/GetEventById/{id}	🔒
GET	/api/Event/GetAllEventsByAttendeeId	🔒
GET	/api/Event/GetAllEventsByMinistryServiceId	🔒
GET	/api/Event/GetAllEventsByChurchId	🔒
POST	/api/Event/CreateEventFromChurch	🔒
POST	/api/Event/CreateEventFromMinistryService	🔒
PATCH	/api/Event/UpdateEvent	🔒
DELETE	/api/Event/DeleteEvent	🔒
1. POST	/api/Event/AddAttendeeToEventId	🔒
POST	/api/Event/RemoveAttendeeFromEventId	🔒

POST /api/Event/AddAttendeeToEventId

Parameters

No parameters

2. Try it out



POST /api/Event/AddAttendeeToEventId

Parameters

No parameters

Request body

3. application/json

4.

```
{  
    "attendeeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",  
    "eventId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"  
}
```

5. Execute

Request body

```
{
  "attendeeId": "ecc3e0e-fafe-4700-aabb-efcf9eab0000",
  "eventId": "2a729ce-eae0-ac4c-aad5-ecf0d0a00000"
}
```

application/json

Execute Clear

Responses

Curl

```
curl -X 'POST' \
  'https://localhost:7282/api/event/addAttendeeToEventId' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJsb2dpbi5jb250ZWdlci5vciIsInN1YiI6ImFkbWluQGxpc3NpZ25lci5vciIsInVzZXJfaWQiOiIxNTMwOTk1ODA5ODIiLCJpYXQiOjE2NjUyNjE4MDAsImV4cCI6MTY2NTg1MjQwM30.' \
  -d '{
    "attendeeId": "ecc3e0e-fafe-4700-aabb-efcf9eab0000",
    "eventId": "2a729ce-eae0-ac4c-aad5-ecf0d0a00000"
  }'
```

Request URL:

<https://localhost:7282/api/event/addAttendeeToEventId>

Server response

Code 500. Details

Response headers

```
Date: Sat, 07 May 2022 10:53:56 GMT
Server: Kestrel
X-Firefox-Spdy: 3.0
```

Description	<p>To add an attendee to event, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the add attendee to eventId tab under the event section in the website API 2. On the right side of the drop down click on “Try it out” 3. Make sure the request body is “application/json” 4. Input the attendee Id which corresponds to the event which the attendee needs to be added to. 5. Execute the command
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Events → Command → AddAttendeeToEventId</p>



Then look at the following files:

1. IEventRepository.cs
2. AddAttendeeToEventCommand.cs
3. AddAttendeeToEventCommandHandler.cs

Infrastructure Layer

src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. EventRepository.cs

10.7.11 Remove Attendee From Event Id

The screenshot shows a list of API endpoints for the 'Event' endpoint. The methods listed are:

- GET /api/Event/GetAllEvents
- GET /api/Event/GetEventById/{id}
- GET /api/Event/GetAllEventsByAttendeeId
- GET /api/Event/GetAllEventsByMinistryServiceId
- GET /api/Event/GetAllEventsByChurchId
- POST /api/Event/CreateEventFromChurch
- POST /api/Event/CreateEventFromMinistryService
- PATCH /api/Event/UpdateEvent
- DELETE /api/Event/DeleteEvent
- POST /api/Event/AddAttendeeToEventId
- 1 POST /api/Event/RemoveAttendeeFromEventId**

Below the list, there is a detailed view for the highlighted method (1):

POST /api/Event/RemoveAttendeeFromEventId

Parameters

No parameters

2 Try it out



POST /api/Event/RemoveAttendeeFromEventId

Parameters

No parameters

Request body

3. application/json

4. {
 "attendeeId": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
 "eventId": "3fa85f64-5717-4562-b3fc-2c963f66afa6"
}

5. Execute

Responses

Curl

```
curl -X 'POST' \  
  'https://localhost:7252/api/Event/RemoveAttendeeFromEventId' \  
  -H 'accept: */*' \  
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJNaWNoYmVsSm9yZGFuIiwianRpIjoiN2FhMzE5YTgtM2Q3ZS00IiwiZW1haWwiOiJ1c2VyQGdtYWlsLmNvbSJ9' \  
  -H 'Content-Type: application/json' \  
  -d '{  
    "attendeeId": "b47fbcc59-f0f9-4380-b053-b398c23a564e",  
    "eventId": "d5343a98-0f2a-4496-87e4-08da2eb94fdb"  
}'
```

Request URL

<https://localhost:7252/api/Event/RemoveAttendeeFromEventId>

Server response

Code	Details
204	Response headers 6. date: Thu, 05 May 2022 17:46:21 GMT server: Kestrel



Description	<p>To remove attendee from an event, refer to the figures above for a sequential visual guide starting with label 1.</p> <ol style="list-style-type: none"> 1. Navigate to the remove attendee from eventId tab under the event section in the website API 2. On the right side of the drop down click on “Try it out” 3. Make sure the request body is “application/json” 4. Input the attendeeId that needs to be removed and the eventId of the event that is being targeted. 5. Execute the function 6. A code 204 response should be generated indicating the removal of the attendee to the event.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Events → Command → RemoveAttendeeFromEventId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IEventRepository.cs 2. RemoveAttendeeFromEventCommand.cs 3. RemoveAttendeeFromEventCommandHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. EventRepository.cs



10.8 Ministry API Functions

10.8.1 Get All Ministry Services

Ministry

1. GET	/api/Ministry/GetAllMinistries	v 🔒
GET	/api/Ministry/GetMinistryById/{id}	v 🔒
GET	/api/Ministry/GetMinistriesByChurchId/{id}	v 🔒
GET	/api/Ministry/GetMinistriesByAttendeeId/{id}	v 🔒
POST	/api/Ministry/CreateMinistry	v 🔒
PATCH	/api/Ministry/UpdateMinistry	v 🔒
DELETE	/api/Ministry/DeleteMinistry	v 🔒
POST	/api/Ministry/AddAttendeeToMinistryServiceById	v 🔒
POST	/api/Ministry/RemoveAttendeeFromMinistryServiceById	v 🔒

Ministry

GET /api/Ministry/GetAllMinistries

Parameters

No parameters

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Example Value | Schema

```
[{"ministryServiceId": "3fa85f64-5717-4562-b2fc-2c96af64ef6e", "name": "string", "address": "string", "eventManager": "string", "phoneNumber": "string", "description": "string"}]
```



GET /api/Ministry/GetAllMinistries

Parameters

No parameters

3. Execute

Responses

Curl

```
curl -x 'xx' \
  https://localhost:7201/api/Ministry/GetAllMinistries \
  -H 'Accept: text/plain' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMwMDAxODQyNDIifQ.eyJpc3MiOiJodHRwczovLmNvbW9kZWQub3Blb25lLmNvbSIsImlhdCI6MTY1NjUyOTIwMSwidXI6ImV4cG9tZSI6MjAxNjUyOTIwMSwiaWF0IjoxNjUyOTIwMSwiaXNzIjoiNzIwLjcyLjEwMC4xMjEiLCJleHAiOjE2MjUyOTIwMSwiaXNzIjoiNzIwLjcyLjEwMC4xMjEiLCJ0eXAiOiJKV1QiLCJub25jZSI6ImFkbWluZC1hbmVzLmNvbSJ9
```

Request URL

<https://localhost:7201/api/Ministry/GetAllMinistries>

Server response

Code Details

200 Response body

```
[{"ministryServiceId": "1ebeded-d3a5-a077-b0ef-8ca0a0000000", "name": "Church Event", "address": "123 Main Street", "eventManager": "John Doe", "phoneNumber": "555-555-5555", "description": "Church function mark event"}, {"ministryServiceId": "7201ca05-408e-4f6c-b6d9-fadcc0000000", "name": "Church Service", "address": "2000 Mountain Hill Drive Portland, Ohio, 45204", "eventManager": "Sarah", "phoneNumber": "444-333-4444", "description": "Regular Church Service"}, {"ministryServiceId": "1ebeded-d3a5-a077-b0ef-8ca0a0000000", "name": "Church Service", "address": "2000 Mountain Hill Drive Portland, Ohio, 45204", "eventManager": "Karen", "phoneNumber": "444-333-4444", "description": "Regular church service"}, {"ministryServiceId": "1ebeded-d3a5-a077-b0ef-8ca0a0000000", "name": "Youth Church Service", "address": "2000 Mountain Hill Drive Portland, Ohio, 45204", "eventManager": "John", "phoneNumber": "444-333-4444", "description": "Youth Church Service"}]
```

4.

Responses

Description	<p>To get all Ministry Services, refer to the figures above for a sequential visual guide starting with label</p> <ol style="list-style-type: none"> 1. Navigate to Get All Ministry Services under Ministry tab on the website API. 2. On the right side of the drop down click on “Try it out” 3. Execute the function 4. 200 Ok Response should be created with the list of all Ministry Services.
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. MinistryController.cs <p>Application Layer</p>



src → Core → Contracts → Persistence
src → Core → Features → Ministry → Query → GetMinistryList

Then look at the following files:

1. IMinistryServiceRepository.cs
2. GetMinistryListQuery.cs
3. GetMinistryListQueryHandler.cs
4. MinistryListVm.cs

10.8.2 Get Ministry By Id

The screenshot shows a REST API documentation interface with two main sections: a list of endpoints and a detailed view of one specific endpoint.

Ministry

GET /api/Ministry/GetAllMinistries

1. GET /api/Ministry/GetMinistryById/{id} (highlighted with a red box)

GET /api/Ministry/GetMinistriesByChurchId/{id}

GET /api/Ministry/GetMinistriesByAttendeeId/{id}

POST /api/Ministry/CreateMinistry

PATCH /api/Ministry/UpdateMinistry

DELETE /api/Ministry/DeleteMinistry

POST /api/Ministry/AddAttendeeToMinistryServiceById

POST /api/Ministry/RemoveAttendeeFromMinistryServiceById

GET /api/Ministry/GetMinistryById/{id}

Parameters

Name **Description**

1 id * required **string(Guid)** (highlighted with a red box)

Try it out (highlighted with a red box)

Responses

Code **Description** **Links**

200 Success No links

Media type **text/plain**

Content Accept header

Example Value **Schema**

```
[{"MinistryGeneralId": "3f9a0f94-5f37-4862-80f1-2e9d0f6efef", "Name": "Worship", "Address": "String", "WebManager": "String", "Administrator": "String", "Description": "String", "ChurchId": "3f9a0f94-5f37-4862-80f1-2e9d0f6efef"}]
```



3.

```

[{"ministryserviceid": "e8b64efc-081a-4e77-b4ff-8ca0d0e0e", "name": "Church Event", "address": "375 Westfall Avenue", "eventManager": "Janet", "phoneNumber": "408-975-8700", "description": "Church insertion Park event"}, {"ministryserviceid": "e8b64efc-081a-4e77-b4ff-8ca0d0e0f", "name": "Church Service", "address": "3870 Service Lane", "eventManager": "Jonathan", "phoneNumber": "404-288-4990", "description": "Regular Church Service"}, {"ministryserviceid": "e8b64efc-081a-4e77-b4ff-8ca0d0e10", "name": "Church Service", "address": "3700 Salem Hill Drive Portland, Ohio, 44084", "eventManager": "Karen", "phoneNumber": "313-229-4612", "description": "Regular Church Service"}, {"ministryserviceid": "e8b64efc-081a-4e77-b4ff-8ca0d0e11", "name": "Church Service", "address": "3700 Salem Hill Drive Portland, Ohio, 44084", "eventManager": "Karen", "phoneNumber": "313-229-4612", "description": "Regular Church Service"}]

```

Description

To get Ministry Services By Id, refer to the figures above for a sequential visual guide starting with label

1. Navigate to Get Ministry Services By Id under Ministry tab on the website API.
2. On the right side of the drop down click on “Try it out” .
3. Input a Ministry Service ID
4. Execute the function
5. 200 Ok Response should be created with the Ministry Service.

List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. MinistryController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features → Ministry → Query → GetMinistryDetail
---------------------------	--



Then look at the following files:

1. IMinistryServiceRepository.cs
2. GetMinistryDetailQuery.cs
3. GetMinistryDetailQueryHandler.cs
4. MinistryDetailVm.cs

10.8.3 Get Ministries By Church Id

The screenshot shows a list of API endpoints for the Ministry controller. The endpoint `/api/Ministry/GetMinistriesByChurchId/{id}` is highlighted with a red box and labeled '1'. The 'Try it out' button for this endpoint is also highlighted with a red box and labeled '2'. The parameter `id` is highlighted with a red box and labeled '3'.

Ministry

- 1. GET `/api/Ministry/GetAllMinistries`
- GET `/api/Ministry/GetMinistryById/{id}`
- 1. GET `/api/Ministry/GetMinistriesByChurchId/{id}`**
- GET `/api/Ministry/GetMinistriesByAttendeeId/{id}`
- POST `/api/Ministry/CreateMinistry`
- PATCH `/api/Ministry/UpdateMinistry`
- DELETE `/api/Ministry/DeleteMinistry`
- POST `/api/Ministry/AddAttendeeToMinistryServiceById`
- POST `/api/Ministry/RemoveAttendeeFromMinistryServiceById`

GET `/api/Ministry/GetMinistriesByChurchId/{id}`

Parameters

Name	Description
3 id <small>* required</small>	<code>id</code>

Responses

Code	Description	Links
200	Success	No links

Example Value Schema

```
{  
    "Ministry": {  
        "id": "1",  
        "name": "Influence",  
        "description": "A ministry focused on spiritual growth.",  
        "leader": "John Doe",  
        "address": "123 Main Street",  
        "phoneNumber": "555-1234",  
        "email": "influence@churchpoint.com"  
    }  
}
```



4

```
curl -X 'GET' \
  'https://localhost:7252/api/Ministry/GetMinistryServicesByChurchId/efc1974-fbfe-4745-b0f6-60d603bec338' \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIaWE1PTENldGNoY29uZmlkZWQ6ZW1haW46b25saW5lIiwidzI6ImV4cG9ydC1pcyIsInR5cGUiOiJodHRwczovL3d3dy5za2V3ZWxpbmUuY29tIiwiaWF0IjoxNjI4MjUyNjEwLCJpYXQiOjE2NjI4MjUyNjEwLCJuYmYiOjE2NjI4MjUyNjEwLCJzZWFyY2hlc2giOjAsImFwcCI6Imh0dHA6Ly9sa2V3ZWxpbmUuY29tL2FwaS8iLCJlbWFpbCI6ImRvY3VtZW50ZWQub3BlcmF0aW9ucy5jb20ifQ=='
```

Responses

Curl

Request URL

Server response

Code Details

200 Response body

```
[{"ministryServiceId": "f5b8266-data-4e77-b0f6-60d603bec338", "name": "Church Event", "address": "1111 Westfall Avenue", "eventCategory": "Jesus", "phoneNumber": "562-495-4729"}]
```

5.

content-type: application/json; charset=utf-8
date: Mon, 04 May 2023 06:55:33 GMT
server: Kestrel
x-aspnet-version: 4.0

Description	To get Ministry Services By Church Id, refer to the figures above for a sequential visual guide starting with label 1. Navigate to Get Ministry Services By Church Id under Ministry tab on the website API. 2. On the right side of the drop down click on “Try it out” . 3. Input a Church ID 4. Execute the function 5. 200 Ok Response should be created with the Ministry Service.
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers API Layer Then look at the following files: 1. MinistryController.cs Application Layer src → Core → Contracts → Persistence src → Core → Features → Ministry → Query → GetAllMinistriesByChurchId Then look at the following files: 1. IMinistryServiceRepository.cs 2. ChurchMinistriesVm.cs 3. GetAllMinistriesByChurchQuery.cs 4. GetAllMinistriesByChurchQueryHandler.cs

Infrastructure Layer
src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:
1. MinistryServiceRepository.cs

10.8.4 Get Ministries By Attendee Id

Ministry

1. GET /api/Ministry/GetAllMinistries

2. GET /api/Ministry/GetMinistryById/{id}

3. GET /api/Ministry/GetMinistriesByChurchId/{id}

1. GET /api/Ministry/GetMinistriesByAttendeeId/{id}

POST /api/Ministry/CreateMinistry

PATCH /api/Ministry/UpdateMinistry

DELETE /api/Ministry/DeleteMinistry

POST /api/Ministry/AddAttendeeToMinistryServiceById

POST /api/Ministry/RemoveAttendeeFromMinistryServiceById

GET /api/Ministry/GetMinistriesByAttendeeId/{id}

Parameters

Name Description

id required string(String) (path) 3

Responses

Code Description Links

200 Success No links

Media type

text/plain

Example Value Schema

```
[{"ministryServiceId": "3fa8f64-5217-4562-bdfc-2e9d3f00efad", "name": "string", "address": "string", "eventManager": "string", "phoneNumber": "string"}]
```



id • required
string(32nd)
(path)

4 Execute Clear

Responses

Curl

```
curl -X 'GET' \
  'https://localhost:7250/api/ministry/MinistryServicesByAttendeeId/0c319688-fa70-476b-a490-1fc5fea33d09' \
  -H 'Accept: application/json' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJuZjIwMjAxMC4xLjAuMSIsInN1YiI6ImFkbWluZWdlciIsInVzZXJfaWQiOiIxIiwiaWF0IjoxNjQyNjE5OTk4LCJleHAiOjE2NzQyNjE5OTk4LCJ0eXAiOiJKV1QiLCJzY29wZWQiOiJodHRwczovL3dyaXRlLmNvbS5jb20iLCJpYXQiOjE2NzQyNjE5OTk4fQ.'
```

Request URL

<https://localhost:7250/api/ministry/MinistryServicesByAttendeeId/0c319688-fa70-476b-a490-1fc5fea33d09>

Server response

Code	Details
200	<p>Response body</p> <pre>[{"MinistryServiceId": "05b5de87-e925-4e74-8ed7-8079206612fa", "Name": "Church Services", "Address": "3000 Walnut Hill Drive Portland, Ohio, 45204", "EventHandler": "Karen", "PhoneNumber": "503-279-8612"}]</pre> <p>Download</p>
Response headers	
<pre>content-type: application/json; charset=utf-8 date: Monday, May 23, 2022 12:09:56 AM server: Kestrel x-frame-options: SAMEORIGIN</pre>	

Description	<p>To get Ministry Services By Attendee Id, refer to the figures above for a sequential visual guide starting with label</p> <ol style="list-style-type: none"> 1. Navigate to Get Ministry Services By AttendeeId under Ministry tab on the website API. 2. On the right side of the drop down click on “Try it out” . 3. Input a Attendee ID 4. Execute the function 5. 200 Ok Response should be created with the Ministry Service(s).
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. MinistryController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Ministry → Query → GetAllMinistriesByAttendeeId</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IMinistryServiceRepository.cs 2. AttendeeMinistriesVm.cs 3. GetAllMinistriesByAttendeeQuery.cs 4. GetAllMinistriesByAttendeeQueryHandler.cs



Infrastructure Layer
src → Infrastructure → ChurchPoint.Persistence → Repositories

Then look at the following files:

1. MinistryServiceRepository.cs

Ministry

GET	/api/Ministry/GetAllMinistries	v	🔒
GET	/api/Ministry/GetMinistryById/{id}	v	🔒
GET	/api/Ministry/GetMinistriesByChurchId/{id}	v	🔒
GET	/api/Ministry/GetMinistriesByAttendeeId/{id}	v	🔒
1 POST	/api/Ministry/CreateMinistry	v	🔒
PATCH	/api/Ministry/UpdateMinistry	v	🔒
DELETE	/api/Ministry/DeleteMinistry	v	🔒
POST	/api/Ministry/AddAttendeeToMinistryServiceById	v	🔒
POST	/api/Ministry/RemoveAttendeeFromMinistryServiceById	v	🔒

POST /api/Ministry/CreateMinistry

Parameters

No parameters

Request body

application/json

Example Value | Schema

3 [{ "churchId": "34567890-1234-5678-9012-345678901234", "name": "String", "address": "String", "phone": "String", "phoneExt": "String", "description": "String" }]

Responses

Code	Description	Links
201	Success	No links

Media type

4 text/plain

Controls Accept header

Example Value | Schema

5 34567890-1234-5678-9012-345678901234



Request body

```
{
  "churchId": "10e02071-e04e-40ff-8332-cba7bd86bf",
  "name": "Testing",
  "address": "123456 test rd",
  "eventManager": "jonathan",
  "phoneNumber": "123-456-4567",
  "description": "Testing to describe what's going on"
}
```

application/json

4. Execute Clear

Responses

Curl

```
curl -X POST \
  https://localhost:7002/api/Ministry/CreateMinistry \
  -H 'accept: application/json' \
  -H 'Content-Type: application/json' \
  -d {
    "churchId": "10e02071-e04e-40ff-8332-cba7bd86bf",
    "name": "Testing",
    "address": "123456 test rd",
    "eventManager": "jonathan",
    "phoneNumber": "123-456-4567",
    "description": "Testing to describe what's going on"
  }
```

Request URL

https://localhost:7002/api/Ministry/CreateMinistry

Server response

Code	Details
201	Response body <code>9A9D9B-ABCD-4E10-8F12-B90026e19C01</code> Download Response headers <code>content-type: application/json; charset=utf-8</code> <code>date: Mon, 04 May 2023 02:35:38 GMT</code> <code>server: Kestrel</code> <code>x-azure-apid: 12</code>

5.

10.8.5 Create Ministry

Description	To create Ministry Services, refer to the figures above for a sequential visual guide starting with label <ol style="list-style-type: none"> 1. Navigate to Create Ministry Service under Ministry tab on the website API. 2. On the right side of the drop down click on “Try it out” 3. Input all needed fields to create a ministry Service (Church Id, Name, address, eventManager, phoneNumber, and description) 4. Execute the function 5. 201 Ok Response should be created with the Ministry Service ID .
List of Code Files	If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers



	<p>API Layer Then look at the following files: 1. MinistryController.cs</p> <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Ministry → Command → CreateMinistry</p> <p>Then look at the following files: 1. IMinistryServiceRepository.cs 2. CreateMinistryCommand.cs 3. CreateMinistryCommandHandler.cs 4. CreateMinistryCommandValidator.cs</p> <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files: 1. MinistryServiceRepository.cs 2. BaseRepository</p>
--	--

10.8.6 Update Ministry

The screenshot shows a list of API endpoints for the Ministry controller. The endpoints are:

- GET /api/Ministry/GetAllMinistries
- GET /api/Ministry/GetMinistryById/{id}
- GET /api/Ministry/GetMinistriesByChurchId/{id}
- GET /api/Ministry/GetMinistriesByAttendeeId/{id}
- POST /api/Ministry/CreateMinistry
- PATCH /api/Ministry/{id} (highlighted with a red box and labeled '1')**
- DELETE /api/Ministry/DeleteMinistry
- POST /api/Ministry/AddAttendeeToMinistryServiceById
- POST /api/Ministry/RemoveAttendeeFromMinistryServiceById

PATCH /api/Ministry/UpdateMinistry

Parameters

No parameters

Request body application/json

Example Value : Schema

```
{
  "ministryServiceId": "3f6ab9f64-5737-4563-b3fc-2c943f66af6c",
  "name": "string",
  "address": "string",
  "eventManager": "string",
  "phoneNumber": "string",
  "description": "string"
}
```

Responses

Code	Description	Links
204	Success	No links

4. **Execute**

Responses

Code

```
curl -X PATCH \
  'https://localhost:7253/api/Ministry/UpdateMinistry' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJsb2dpbi5jb20udmlyZWFzdCIsInN1YiI6ImFkbWluLWRhLWxvY2FsaG9tIiwidXNlcm5hbWUiOiJhZG1pbiIsInByZWZhbWUiOiJhZG1pbiIsInBhY2thZ2VzIjoiZWxvY2FsaG9tLWRhLWxvY2FsaG9tIn0.' \
  -d '{
    "ministryServiceId": "3f6ab9f64-5737-4563-b3fc-2c943f66af6c",
    "name": "New Meeting",
    "address": "789 street way",
    "eventManager": "John Doe",
    "phoneNumber": "123-456-7890",
    "description": "New meeting"
}'
```

Request URL

5. **Server response**

Code	Details
204	Response headers date: Mon, 01 May 2023 00:44:34 GMT server: Apache x-frame-options: SAMEORIGIN

Description	To Update Ministry Services, refer to the figures above for a sequential visual guide starting with label 1. Navigate to Update Ministry Service under Ministry tab on the website API. 2. On the right side of the drop down click on “Try it out”
--------------------	---



	<ol style="list-style-type: none"> 3. Update any fields to update a ministry Service using the ministry service id (Name, address, eventManager, phoneNumber, and description) 4. Execute the function 5. 204 Success Response should be created with response of time .
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. MinistryController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Ministry → Command → UpdateMinistry</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IMinistryServiceRepository.cs 2. UpdateMinistryCommand.cs 3. UpdateMinistryCommandHandler.cs 4. UpdateMinistryCommandValidator <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. MinistryServiceRepository.cs 2. BaseRepository



10.8.7 Delete Ministry

Ministry

GET	/api/Ministry/GetAllMinistries	v
GET	/api/Ministry/GetMinistryById/{id}	v
GET	/api/Ministry/GetMinistriesByChurchId/{id}	v
GET	/api/Ministry/GetMinistriesByAttendeeId/{id}	v
POST	/api/Ministry/CreateMinistry	v
PATCH	/api/Ministry/UpdateMinistry	v
1 DELETE	/api/Ministry/DeleteMinistry	v
POST	/api/Ministry/AddAttendeeToMinistryServiceById	v
POST	/api/Ministry/RemoveAttendeeFromMinistryServiceById	v

DELETE /api/Ministry/DeleteMinistry

Parameters

Name	Description
Id	string 3 (query)

Responses

Code	Description	Links
204	Success	No links
404	Not Found	No links

Media type:

Example Value:

```
{ "type": "string", "title": "string", "status": "string", "detail": "string", "instance": "string", "additionalProp1": "string", "additionalProp2": "string", "additionalProp3": "string" }
```



Description	<p>To Delete Ministry Services, refer to the figures above for a sequential visual guide starting with label</p> <ol style="list-style-type: none"> 1. Navigate to Delete Ministry Service under Ministry tab on the website API. 2. On the right side of the drop down click on “Try it out” 3. Input a ministry service id 4. Execute the function 5. 204 Success Response should be created with response of time .
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p>

Then look at the following files:

1. MinistryServiceRepository.cs
2. BaseRepository

10.8.8 Add Attendee to Ministry Service

Ministry

- GET** /api/Ministry/GetAllMinistries
- GET** /api/Ministry/GetMinistryById/{id}
- GET** /api/Ministry/GetMinistriesByChurchId/{id}
- GET** /api/Ministry/GetMinistriesByAttendeeId/{id}
- POST** /api/Ministry/CreateMinistry
- PATCH** /api/Ministry/UpdateMinistry
- DELETE** /api/Ministry/DeleteMinistry
- 1** **POST** /api/Ministry/AddAttendeeToMinistryServiceById
- POST** /api/Ministry/RemoveAttendeeFromMinistryServiceById

POST /api/Ministry/AddAttendeeToMinistryServiceById

Parameters

No parameters

Request body

application/json

Example Value | Schema

```
3 {"AttendeeId": "3fabbf64-5737-45d2-b3fc-3c9d166efef", "MinistryServiceId": "3fabbf64-5737-45d2-b3fc-3c9d166efef"}
```

Responses

Code	Description	Links
204	Success	No links



The screenshot shows the ChurchPoint API documentation interface. At the top, there is a JSON payload for the 'AddAttendeeToMinistryServiceById' endpoint. Below it, there is a red box around the 'Execute' button. The 'Responses' section contains a 'Curl' command and a 'Request URL' (https://localhost:7252/api/Ministry/AddAttendeeToMinistryServiceById). The 'Server response' section shows a 204 status code with a red box around it. The response headers are listed as follows:

```

204
5.
  Response headers
  date: Mon, 02 May 2022 05:34:29 GMT
  server: IIS/10.0
  x-azure-apd: 0

```

Description	<p>To add an attendee to Ministry Service By Id, refer to the figures above for a sequential visual guide starting with label</p> <ol style="list-style-type: none"> 1. Navigate to Add Attendee To MinistryService By Id under Ministry tab on the website API. 2. On the right side of the drop down click on “Try it out” . 3. Input a Attendee ID and Ministry Service Id 4. Execute the function 5. 204 Success Response should be created with response of time .
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. MinistryController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Ministry → Command → AddAttendeeToMinistryService</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IMinistryServiceRepository.cs



	<p>2. AddAttendeeToMinistryServiceCommand.cs 3. AddAttendeeToMinistryServiceCommandHandler.cs</p> <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. MinistryServiceRepository.cs
--	--

10.8.9 Remove Attendee From Ministry Service

The screenshot shows a list of API endpoints for the Ministry service. The endpoints are color-coded by HTTP method: GET (blue), POST (green), PATCH (light green), DELETE (red), and another POST (light green). The highlighted endpoint, POST /api/Ministry/RemoveAttendeeFromMinistryServiceById, is also outlined with a red box.

Ministry	
GET	/api/Ministry/GetAllMinistries
GET	/api/Ministry/GetMinistryById/{id}
GET	/api/Ministry/GetMinistriesByChurchId/{id}
GET	/api/Ministry/GetMinistriesByAttendeeId/{id}
POST	/api/Ministry/CreateMinistry
PATCH	/api/Ministry/UpdateMinistry
DELETE	/api/Ministry/DeleteMinistry
POST	/api/Ministry/AddAttendeeToMinistryServiceById
POST	/api/Ministry/RemoveAttendeeFromMinistryServiceById

POST /api/Ministry/RemoveAttendeeFromMinistryServiceById

Parameters

No parameters

Request body

application/json

Example Value | Schema

3. { "attendeeId": "00000003-51d3-4abe-ac4d-c0f0b1bd1d0f", "ministryServiceId": "f625e9be-1106-4881-903e-22753f00e0ff" }

Responses

Code	Description	Links
204	Success	No links
404	Not Found	No links

Media type

text/plain

Example Value | Schema

{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "additionalProp1": "string", "additionalProp2": "string", "additionalProp3": "string" }

4. Execute

Responses

Curl

```
curl -X POST \
  'https://localhost:7252/api/Ministry/RemoveAttendeeFromMinistryServiceById' \
  -H 'accept: */*' \
  -H 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjMwMDA2MDIyMSIsInN1YiI6IjUwMDAwMDAwMDAwMCIsImF1ZGUiOiJsb2dpbiIsIiwiZXhwIjoxNjIwOTQwODk4fQ.' \
  -H 'Content-Type: application/json' \
  -d '{ "attendeeId": "00000003-51d3-4abe-ac4d-c0f0b1bd1d0f", "ministryServiceId": "f625e9be-1106-4881-903e-22753f00e0ff" }'
```

Request URL

https://localhost:7252/api/Ministry/RemoveAttendeeFromMinistryServiceById

Server response

Code Details

204

Response headers

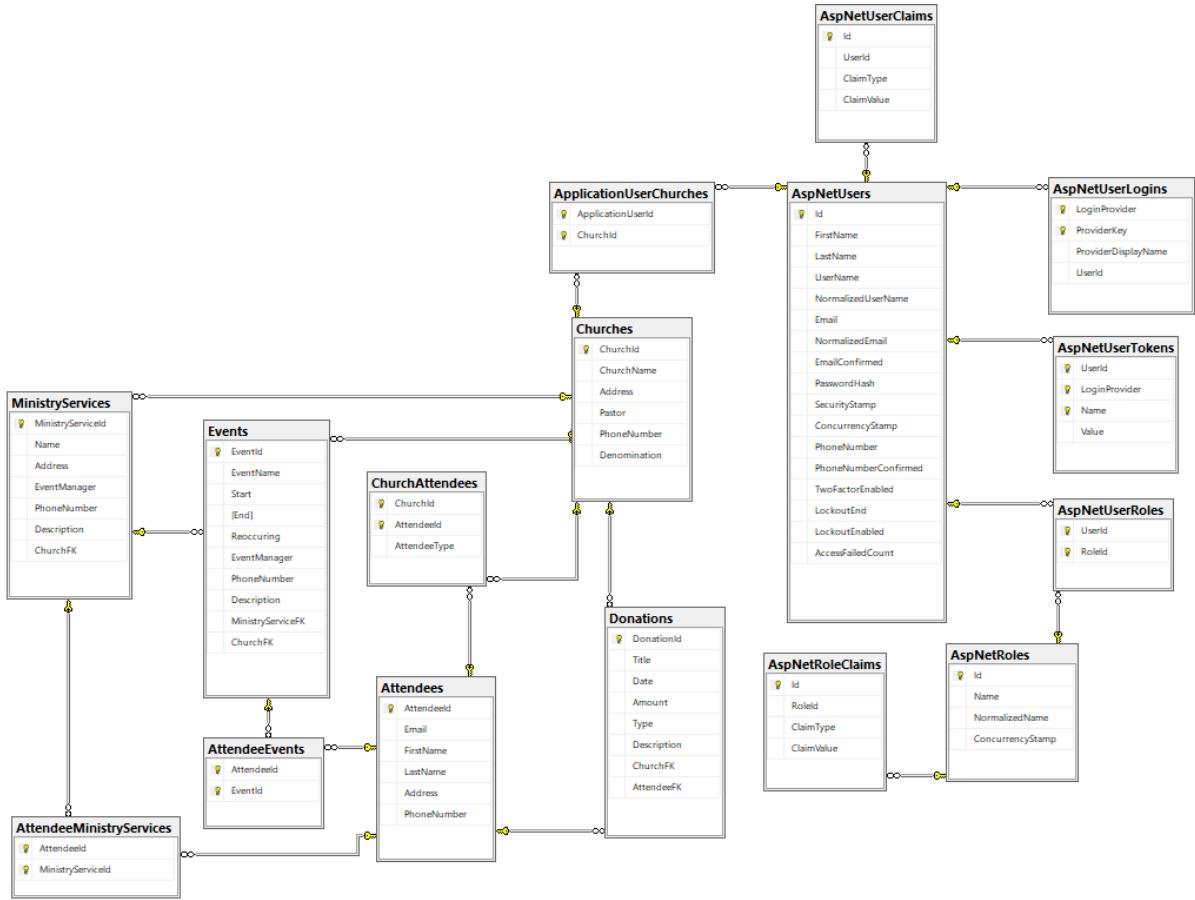
```
Date: Mon, 01 May 2022 00:26:58 GMT
Server: Kestrel
X-Friendly-App: 10
```



Description	<p>To remove an attendee to Ministry Service By Id, refer to the figures above for a sequential visual guide starting with label</p> <ol style="list-style-type: none"> 1. Navigate to remove Attendee To MinistryService By Id under Ministry tab on the website API. 2. On the right side of the drop down click on “Try it out” . 3. Input a Attendee ID and Ministry Service Id 4. Execute the function 5. 204 Success Response should be created with response of time .
List of Code Files	<p>If there are any errors go to the Solution Explorer and Navigate to src → ChurchPoint.API → Controllers</p> <p>API Layer Then look at the following files:</p> <ol style="list-style-type: none"> 1. MinistryController.cs <p>Application Layer src → Core → Contracts → Persistence src → Core → Features → Ministry → Command → RemoveAttendeeFromMinistryService</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. IMinistryServiceRepository.cs 2. RemoveAttendeeFromMinistryService.cs 3. RemoveAttendeeFromMinistryServiceCommandHandler.cs <p>Infrastructure Layer src → Infrastructure → ChurchPoint.Persistence → Repositories</p> <p>Then look at the following files:</p> <ol style="list-style-type: none"> 1. MinistryServiceRepository.cs



11 Database



Above is the database schema produced by SQL Management Studio. Our Database design has 16 tables, 10 of which comprise non-framework based relationships in the database.

Maintenance Manual Approval

System testing is the process whereby complete, integrated software application is tested and is intended to evaluate and verify the system's compliance with the specified requirements. The System Test document contains the descriptions of the system tests used.

Product Owner

Pacifico Jun

Former Verizon Software Analyst / Entrepreneur / Pastor
Allegiant Church

The Maintenance Manual Document prepared by Allegiant.Dev

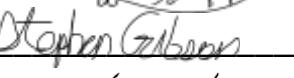
Plx Mapanao



Nathan Hidalgo



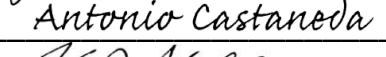
Stephen Gibson



Jonathan Lopez



Antonio Castaneda



Keith Valdes



Emanuel Onofre



Date 5/11/2022

Technical Consultant



Wilver Linogon

Allegiant Church Tech
Consultant

