**Техническая документация Музея**

**Содержание**

[Описание классов 5](#_Toc124111857)

[Пространство имен Admin.PHP 5](#_Toc124111858)

[QueriesToPHP.cs 5](#_Toc124111859)

[HallQueries.cs 5](#_Toc124111860)

[Пространство имен Admin.Auth 7](#_Toc124111861)

[AuthPanel.cs 7](#_Toc124111862)

[VersionText.cs 7](#_Toc124111863)

[ILoggable.cs 7](#_Toc124111864)

[Logger.cs 7](#_Toc124111865)

[IFieldsProvider.cs 8](#_Toc124111866)

[LoginFieldsProvider.cs 8](#_Toc124111867)

[RegistrationFieldsProvider.cs 8](#_Toc124111868)

[PasswordSaver.cs 8](#_Toc124111869)

[Login.cs 8](#_Toc124111870)

[Registration.cs 8](#_Toc124111871)

[Пространство имен Admin.View 10](#_Toc124111872)

[HallLister.cs 10](#_Toc124111873)

[HallPreviewResizer.cs 10](#_Toc124111874)

[HallViewer.cs 10](#_Toc124111875)

[TilesDrawer.cs 11](#_Toc124111876)

[Пространство имен Admin.Edit 12](#_Toc124111877)

[ProgressSaver.cs 12](#_Toc124111878)

[EditCursor.cs 12](#_Toc124111879)

[ToolSelector.cs 12](#_Toc124111880)

[RubberBrush.cs 12](#_Toc124111881)

[PaintBrush.cs 13](#_Toc124111882)

[EditBrush.cs 13](#_Toc124111883)

[EditMedia.cs 13](#_Toc124111884)

[EditInfoBox.cs 13](#_Toc124111885)

[InfoPart.cs 13](#_Toc124111886)

[EditDecoration.cs 13](#_Toc124111887)

[HallEditor.cs 13](#_Toc124111888)

[Пространство имен Admin.New 15](#_Toc124111889)

[HallCreationDate.cs 15](#_Toc124111890)

[HallCreation.cs 15](#_Toc124111891)

[Пространство имен Admin.UsersManagement 16](#_Toc124111892)

[UsersParser.cs 16](#_Toc124111893)

[UsersSpawner.cs 16](#_Toc124111894)

[UserContent.cs 16](#_Toc124111895)

[Пространство имен Admin.Utility 17](#_Toc124111896)

[JsonHelper.cs 17](#_Toc124111897)

[RaycastUtilities.cs 17](#_Toc124111898)

[AccessLevel.cs 17](#_Toc124111899)

[User.cs 17](#_Toc124111900)

[Hall.cs 17](#_Toc124111901)

[HallContent.cs 17](#_Toc124111902)

[PanelChanger.cs 17](#_Toc124111903)

[Painting.cs 18](#_Toc124111904)

[Video.cs 18](#_Toc124111905)

[VideoFrame.cs 18](#_Toc124111906)

[Без пространства имен 19](#_Toc124111907)

[InfoController.cs 19](#_Toc124111908)

[MuseumPreviewSize.cs 19](#_Toc124111909)

[Tile.cs 19](#_Toc124111910)

[Внешние зависимости, как их заменять 20](#_Toc124111911)

[Плагин Automatic UI Anchoring 20](#_Toc124111912)

[Плагин WebGLCopyAndPaste 20](#_Toc124111913)

[Видео из GitHub Pages 20](#_Toc124111914)

[PHP скрипты для БД 20](#_Toc124111915)

[База данных MySQL 20](#_Toc124111916)

[Перенос проекта 21](#_Toc124111917)

[Компиляция проекта 21](#_Toc124111918)

[Перенос веб-приложения администраторской части 21](#_Toc124111919)

[Перенос веб-приложения клиентской части 21](#_Toc124111920)

[Обновление проекта на новую версию Unity 21](#_Toc124111921)

[Принятые решения и ограничения 22](#_Toc124111922)

[Почему PHP? 22](#_Toc124111923)

[Почему MySQL? 22](#_Toc124111924)

[Почему хостинг Awardspace? 22](#_Toc124111925)

[Почему видео из GitHub Pages? 22](#_Toc124111926)

# Описание классов

## Пространство имен Admin.PHP

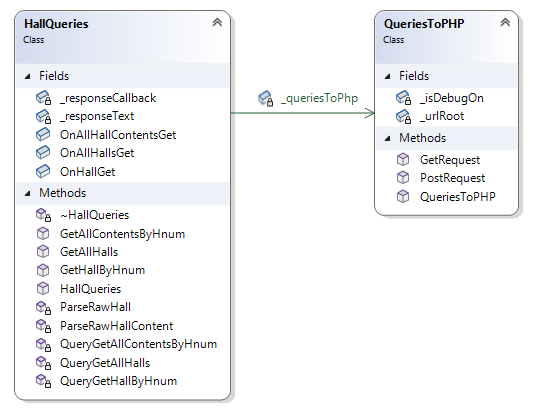


Рис. 1. Диаграмма классов

### QueriesToPHP.cs

Класс, отвечающий за GET/POST запросы к PHP скриптам. Запросы выполняются внутри IEnumerator методах (корутинах), которые принимают название PHP скрипта на сервере, возвращают – обратный вызов с текстом-ответом из скрипта. Для использования класса нужно сконструировать его C# объект, обращаться к методам через объект.

### HallQueries.cs

Класс, отвечающий за ООП запросы к базе данных, относящиеся к залам музея. Имеет 3 публичных контракта-корутины – GetHallByHnum (получить зал по его ID), GetAllHalls (получить все залы), GetAllContentsByHnum (получить все наполнение зала по его ID). При успешном выполнении этих методов, вызываются обратные вызовы OnHallGet, OnAllHallsGet, OnAllHallContentsGet, которые передают объекты или списки объектов. Для использования класса нужно сконструировать его C# объект, обращаться к методам через объект.

## Пространство имен Admin.Auth

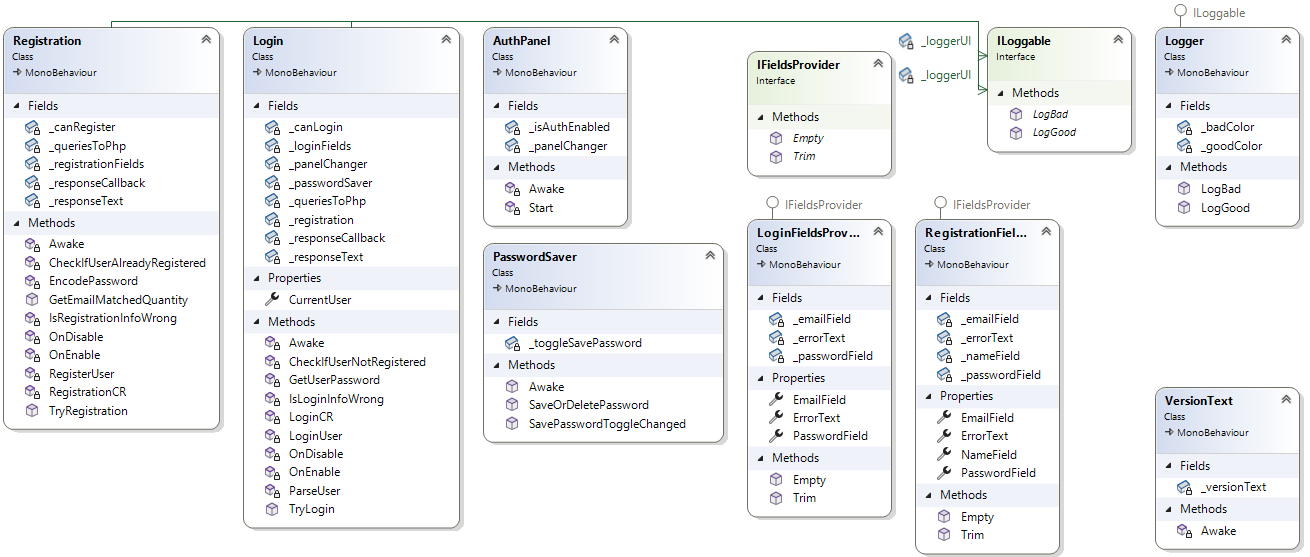


Рис. 2. Диаграмма классов

### AuthPanel.cs

MonoBehaviour, который устанавливает полноэкранный режим на старте, а также включает панель авторизации (либо пропускает ее при выключенном флаге \_isAuthEnabled).

### VersionText.cs

MonoBehaviour, который на старте приложения отображает версию приложения в виде текста в интерфейсе.

### ILoggable.cs

Интерфейс, содержащий методы LogGood, LogBad для логгирования хорошего или плохого сообщения.

### Logger.cs

MonoBehaviour, заключен контракт с интерфейсом ILoggable, дает в использование публичные методы для логгирования в пользовательском интерфейсе.

### IFieldsProvider.cs

Интерфейс, содержащий методы Empty (очистить содержимое поля), Trim (обрезать поле с двух сторон).

### LoginFieldsProvider.cs

MonoBehaviour, заключен контракт с интерфейсом IFieldsProvider, дает в использование публичные методы для манипуляции с полями ввода текста, используемые при логине пользователя.

### RegistrationFieldsProvider.cs

MonoBehaviour, заключен контракт с интерфейсом IFieldsProvider, дает в использование публичные методы для манипуляции с полями ввода текста, используемые при регистрации пользователя.

### PasswordSaver.cs

MonoBehaviour, который занимается логикой сохранения или удаления пароля, если отмечена галочка «Запомнить пароль». Использует PlayerPrefs для сохранения (максимальный вес сохраняемых данных – 1МБ, сохраняются они в IndexedDB браузера).

### Login.cs

MonoBehaviour, который занимается логикой авторизации пользователя, также получает пользователя из базы данных и запоминает его в публичное авто-свойство CurrentUser, которое используется множеством других классов.

### Registration.cs

MonoBehaviour, который занимается логикой регистрации пользователя.

## Пространство имен Admin.View

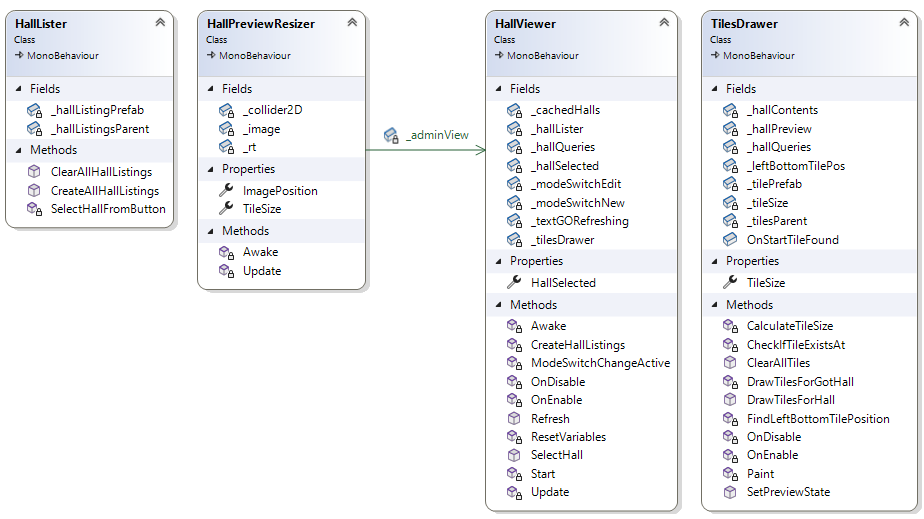


Рис. 3. Диаграмма классов

### HallLister.cs

MonoBehaviour, отвечающий за создание и удаление интерфейсных элементов – списка доступных залов.

### HallPreviewResizer.cs

MonoBehaviour, отвечающий за изменение размера предварительного просмотра у выбранного 2D плана зала.

### HallViewer.cs

MonoBehaviour, отвечающий за выбор зала. Хранит в себе публичное свойство HallSelected, которое используется множеством других классов.

### TilesDrawer.cs

MonoBehaviour, отвечающий за получение содержимого выбранного зала, его отрисовку в интерфейсе, поверх 2D плана.

## Пространство имен Admin.Edit

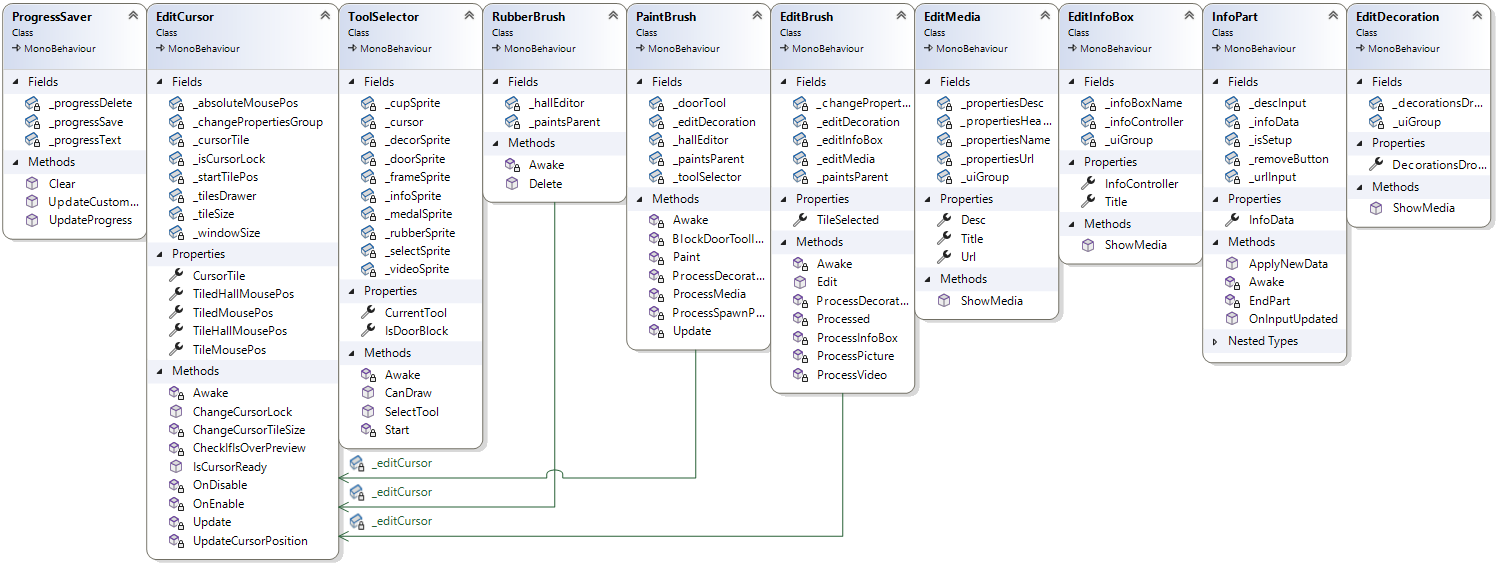


Рис. 4. Диаграмма классов

### ProgressSaver.cs

MonoBehaviour, отвечающий за визуальное отображение в тексте прогресса сохранения настроек зала.

### EditCursor.cs

MonoBehaviour, отвечающий за просчет положения и отображение курсора в интерфейсе.

### ToolSelector.cs

MonoBehaviour, отвечающий за выбор инструмента из палитры, а также изменение иконки курсора.

### RubberBrush.cs

MonoBehaviour, отвечающий за удаление иконки на месте курсора при выбранном инструменте резинки.

### PaintBrush.cs

MonoBehaviour, отвечающий за рисование выбранной иконки из палитры на 2D плане зала, а также установление начального состояния нарисованного предмета музея.

### EditBrush.cs

MonoBehaviour, отвечающий за редактирование (изменение) предмета на месте курсора, его обработка в зависимости от его типа.

### EditMedia.cs

MonoBehaviour, отвечающий за отображение настроек медиа (фото, видео) на пользовательском интерфейсе.

### EditInfoBox.cs

MonoBehaviour, отвечающий за отображение настроек информационного стенда на пользовательском интерфейсе.

### InfoPart.cs

MonoBehaviour, отвечающий за часть информации для информационного стенда – проверка ввода, сохранение, удаление. Также содержит структуру InfoPartData, описывающую ссылка на картинку и описание.

### EditDecoration.cs

MonoBehaviour, отвечающий за отображение настроек декорации на пользовательском интерфейсе.

### HallEditor.cs

MonoBehaviour, отвечающий за редактирование настроек зала – даты открытия/закрытие, название зала, его доступность, удаление. Также хранит в себе массив, являющийся 2D планом содержимого зала. В нем происходят все взаимодействия с базой данных – сохранение настроек зала, содержимого, их удаление.

## Пространство имен Admin.New

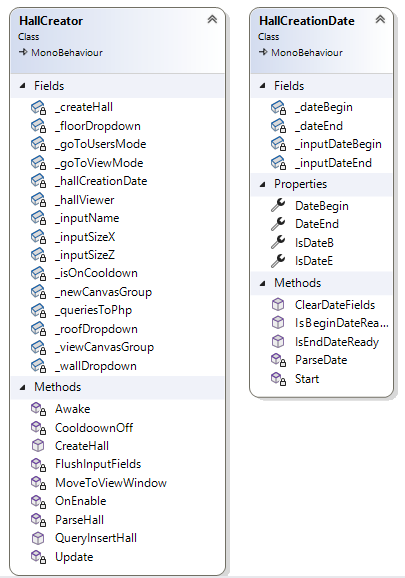


Рис. 5. Диаграмма классов

### HallCreationDate.cs

MonoBehaviour, отвечающий за ввод даты открытия и закрытия зала пользователем.

### HallCreation.cs

MonoBehaviour, отвечающий за парсинг и валидацию введенных данных нового зала, отправляет запрос к базе данных на его создание.

## Пространство имен Admin.UsersManagement

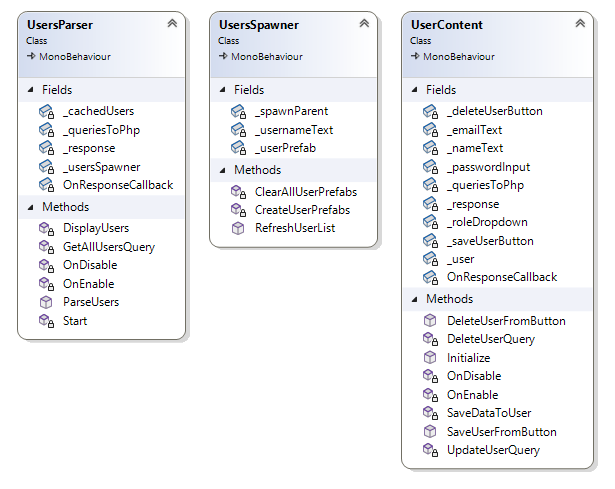


Рис. 6. Диаграмма классов

### UsersParser.cs

MonoBehaviour, отвечающий за получение всех пользователей из БД, кеширует их в список пользователей.

### UsersSpawner.cs

MonoBehaviour, отвечающий за создание всех кешированных пользователей на пользовательском интерфейсе.

### UserContent.cs

MonoBehaviour, отвечающий за представление пользователя в интерфейсе – отображение его полей, сохранение изменений у пользователя в базе данных и удаление пользователя.

## Пространство имен Admin.Utility

### JsonHelper.cs

Класс, отвечающий за сериализацию и десериализацию массивов в JSON и обратно. Доступный по умолчанию в Unity класс JsonUtility не способен корректно это делать.

### RaycastUtilities.cs

Класс, отвечающий за получение элемента интерфейса, на который указывает курсор.

### AccessLevel.cs

Перечисление, описывающее роли, возможные у пользователя.

### User.cs

Структура данных, описывающая пользователя. Содержит в себе AccessLevel.

### Hall.cs

Структура данных, описывающая зал Музея, совпадает с структурой соответствующей таблицы в базе данных.

### HallContent.cs

Структура данных, описывающая наполнение зала Музея, совпадает с структурой соответствующей таблицы в базе данных.

### PanelChanger.cs

Содержит перечисление Panel, отвечающее за UI-панели в приложении, а также логику переключения между этими панелями.

### Painting.cs

MonoBehaviour, отвечающий за загрузку и отображение картинки из Интернета по ссылке.

### Video.cs

MonoBehaviour, отвечающий за загрузку и воспроизведение видео в текстуре из Интернета (GitHub Pages) по ссылке.

### VideoFrame.cs

MonoBehaviour, отвечающий за загрузку и воспроизведение видео из Интернета (GitHub Pages) по ссылке при взаимодействии с пользователем.

## Без пространства имен

### InfoController.cs

MonoBehaviour, отвечающий за управление информационным стендом: добавление и удаление его частей, сериализация всей информации в JSON для последующего сохранения в БД.

### MuseumPreviewSize.cs

MonoBehaviour, отвечающий за изменение размера предварительного просмотра плана 2D музея.

### Tile.cs

MonoBehaviour, отвечающий за визуальное представление иконки на 2D плане Музея.

# Внешние зависимости, как их заменять

## Плагин Automatic UI Anchoring

## Плагин WebGLCopyAndPaste

## Видео из GitHub Pages

## PHP скрипты для БД

## База данных MySQL

# Перенос проекта

## Компиляция проекта

## Перенос веб-приложения администраторской части

## Перенос веб-приложения клиентской части

## Обновление проекта на новую версию Unity

# Принятые решения и ограничения

## Почему PHP?

## Почему MySQL?

## Почему хостинг Awardspace?

## Почему видео из GitHub Pages?