**Техническая документация Музея**

**Содержание**

[Описание классов 3](#_Toc124106742)

[Пространство имен Admin.PHP 3](#_Toc124106743)

[QueriesToPHP.cs 3](#_Toc124106744)

[HallQueries.cs 3](#_Toc124106745)

[Пространство имен Admin.Auth 4](#_Toc124106746)

[AuthPanel.cs 4](#_Toc124106747)

[VersionText.cs 4](#_Toc124106748)

[ILoggable.cs 5](#_Toc124106749)

[Logger.cs 5](#_Toc124106750)

[IFieldsProvider.cs 5](#_Toc124106751)

[LoginFieldsProvider.cs 5](#_Toc124106752)

[RegistrationFieldsProvider.cs 5](#_Toc124106753)

[PasswordSaver.cs 5](#_Toc124106754)

[Login.cs 6](#_Toc124106755)

[Registration.cs 6](#_Toc124106756)

[Пространство имен Admin.View 6](#_Toc124106757)

[Пространство имен Admin.Edit 6](#_Toc124106758)

[Пространство имен Admin.New 7](#_Toc124106759)

[Пространство имен Admin.UsersManagement 8](#_Toc124106760)

[Пространство имен Admin.Utility 8](#_Toc124106761)

[Внешние зависимости, как их заменять 10](#_Toc124106762)

[Плагин Automatic UI Anchoring 10](#_Toc124106763)

[Плагин WebGLCopyAndPaste 10](#_Toc124106764)

[Видео из GitHub Pages 10](#_Toc124106765)

[PHP скрипты для БД 10](#_Toc124106766)

[База данных MySQL 10](#_Toc124106767)

[Перенос проекта 11](#_Toc124106768)

[Компиляция проекта 11](#_Toc124106769)

[Перенос веб-приложения администраторской части 11](#_Toc124106770)

[Перенос веб-приложения клиентской части 11](#_Toc124106771)

[Обновление проекта на новую версию Unity 11](#_Toc124106772)

[Принятые решения и ограничения 12](#_Toc124106773)

[Почему PHP? 12](#_Toc124106774)

[Почему MySQL? 12](#_Toc124106775)

[Почему хостинг Awardspace? 12](#_Toc124106776)

[Почему видео из GitHub Pages? 12](#_Toc124106777)

# Описание классов

## Пространство имен Admin.PHP

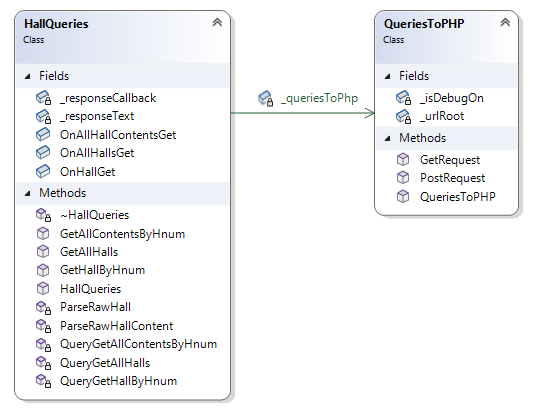


Рис. 1. Диаграмма классов

### 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

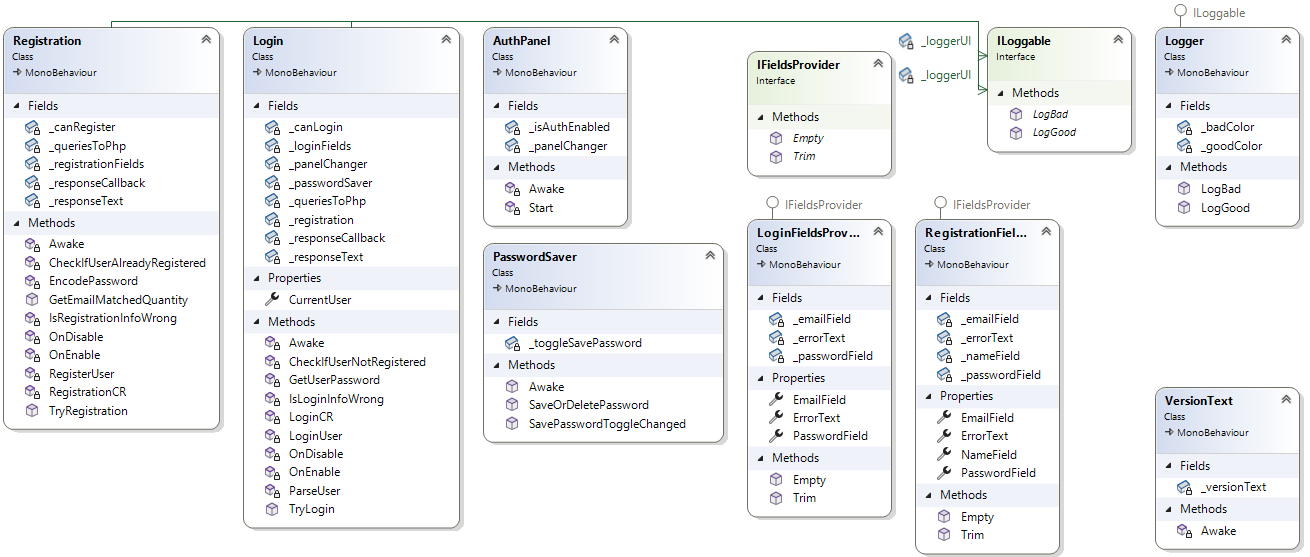


Рис. 2. Диаграмма классов

### 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

## Пространство имен Admin.Edit



Рис. 4. Диаграмма классов

## Пространство имен Admin.New



Рис. 5. Диаграмма классов

## Пространство имен Admin.UsersManagement



Рис. 6. Диаграмма классов

## Пространство имен Admin.Utility



Рис. 7. Диаграмма классов

# 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?