Пермский филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский университет

«Высшая школа экономики»

Факультет экономики, менеджмента и бизнес-информатики

Образовательная программа бакалавриата «Программная инженерия»

**ОТЧЕТ  
по технологической практике**

Выполнил студент группы ПИ-18-2

Чепоков Елизар Сергеевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

Проверил:

Руководитель практики

к.т.н, доцент кафедры

информационных технологий в бизнесе

Кычкин Алексей Владимирович

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

**Пермь, 2020**

**Аннотация**

Автор: Чепоков Елизар Сергеевич, студент второго курса образовательной программы «Программная инженерия».

Тема практической работы: «Реализация web-приложения для подключения к платформе EMS INSYTE с помощью MQTT протокола».

Руководитель: Кычкин Алексей Владимирович, доцент кафедры информационных технологий в бизнесе.

Данная практическая работа посвящена разработке Web-приложения для подключения к платформе EMS INSYTE реализованной нa языках C++, JavaScript и языка разметки Html по средствам среды Microsoft Visual Studio.

Работа включает 193 страниц формата А4, из них в основной части 30 страниц.

Основная часть работы включает в себя 9 иллюстраций и 7 таблиц.

Библиографический список состоит из 6 публикаций.

Работа включает в себя 5 приложений.

# Оглавление

[Оглавление 3](#_Toc41936766)

[Введение 5](#_Toc41936767)

[Глава 1. Анализ 6](#_Toc41936768)

[1.1. Изучение технологии MQTT протокола 6](#_Toc41936769)

[1.2. Обзор брокеров и клиентов 7](#_Toc41936770)

[1.3. Требования к разрабатываемой программе 10](#_Toc41936771)

[Глава 2. Проектирование 11](#_Toc41936772)

[2.1. Проектирование алгоритмов 11](#_Toc41936773)

[2.1.1. Создание диаграмм взаимодействия информационной системы 17](#_Toc41936774)

[2.2. Проектирование интерфейса 18](#_Toc41936775)

[Глава 3. Разработка и тестирование. 19](#_Toc41936778)

[3.1. Выбор инструментов разработки 19](#_Toc41936785)

[3.2. Разработка интерфейса. 19](#_Toc41936786)

[3.3. Кодирование. 23](#_Toc41936787)

[3.4. Тестирование программного продукта. 27](#_Toc41936788)

[3.4.1. Планирование тестирования. 27](#_Toc41936789)

[3.4.2. Планирование процедур контроля качества. 28](#_Toc41936790)

[3.4.3. Интеграционное тестирование. 28](#_Toc41936791)

[3.4.4. Системное тестирование. 28](#_Toc41936792)

[Заключение 29](#_Toc41936793)

[Список сокращений и условных обозначений 30](#_Toc41936794)

[Библиографический список 31](#_Toc41936795)

[Приложение А 32](#_Toc41936796)

[Приложение Б 36](#_Toc41936797)

[Приложение В 37](#_Toc41936798)

[Приложение Г 191](#_Toc41936799)

[Приложение Д 193](#_Toc41936800)

# Введение

Технологическая практика заключается в создании Web-приложения для подключения к платформе «EMS INSYTE» с помощью MQTT протокола. Приложение служит расширением возможностей использования системы пользователем, при помощи интернет ресурсов. Так же такой подход позволит расширить аудиторию и удобство в использовании системы «Умный дом».

В связи с тем, что темой курсовой работы является разработка программы для контроллера «Arduino», то темой данной технологической практики будет разработка клиента для подключения к платформе «EMS INSYTE», так как перенос программы для контроллера «Arduino» на web-приложение является не нужным из-за отсутствия области применения.

Web-клиенты для подключения к удаленному брокеру уже существуют, но для большей безопасности пользователя, каждой компании рекомендуемо создавать свои аналоги с закрытым кодом, так как такие системы менее подвержены взлому.

# Анализ

В данной главе будут рассмотрены особенности работы и взаимодействия клиента и MQTT брокера, а также особенности MQTT протокола. Будут изучены аналоги брокеров и клиентов для работы с брокерами, выявлены их преимущества и недостатки. Результатом первой главы будут требования к программному продукту, а также диаграмма прецедентов.

## Изучение технологии MQTT протокола

Для связи контроллера с облачным брокером, потребуется научить контроллер выходить в сеть и сообщать о любых проблемах, возникших при подключении. Для данной задачи требуется выбрать сетевой протокол, который будет задавать направление в подключении.

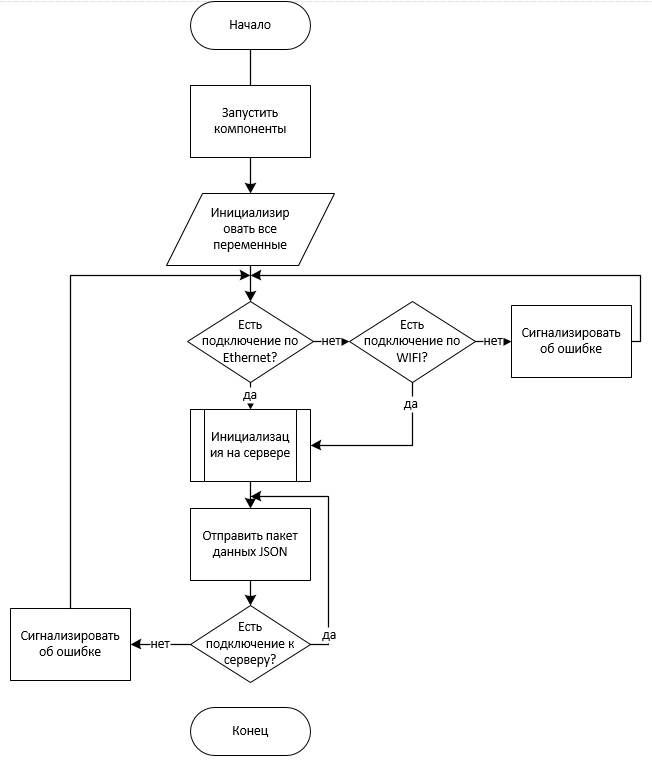
Сетевой протокол – это набор правил и действий в очередном порядке, позволяющий осуществить соединение и обмен данными между включенными в сеть устройствами. Минусом большинства протоколов является постоянная поддержка связи.

MQTT протокол – это упрощённый сетевой протокол работающий по принципу издатель-подписчик. Протокол этого типа ориентируется на простоту использования, легкую встраиваемость в любую систему и работу в условиях постоянной потери связи.

Так как планируется создать контроллер, который будет прост в создании и настройке и подключаться периодически, а не все время, то использоваться будет MQTT протокол.

Для правильной работы контроллера протокол должен выполнять следующие функции:

* + 1. Подключение к брокеру;
    2. Оповещение об отсутствии подключения;
    3. Подписка на топики;
    4. Отправка данных;
    5. Перепроверка подключения.

На рисунке 1.1 представлена упрощенная версия действий протокола для подключения к брокеру.

**Рисунок 1.1 – Протокол MQTT**

## Обзор брокеров и клиентов

Брокер является посредником для передачи данных между пользователем и устройством, такая система позволяет управлять предметами умного дома из любого места, при наличии подключения к сети «Интернет».

Целью данной курсовой работы является реализация подключения контроллера устройства, к удаленному брокеру «EMS INSYTE». Исходя из этого брокером будет являться система «EMS INSYTE»**.**

Написание отдельного брокера является крайне сложным и долгим проектом и качество результата будет низким и дорогим, так как будет требовать вложений для выхода в online. По этой причине данная курсовая работа не будет рассматривать этот вариант.

EMS INSYTE – автоматизированная система по управлению техническими устройствами, созданная Пермской компанией.

Данная система создана преимущественно для модерирования умным домом на практике. Так как система является коммерческим продуктом, она не приспособлена для легкого взаимодействия и обучения, и для отправки сообщения нужно точно знать куда и что отправлять. Данный подход обеспечивает безопасность умного дома и передаваемых данных, а так же сложность взаимодействия с ним.

Из клиентов для подключения к брокерам, самыми используемыми являются:

* MQTTspy;
* OpenHAB;
* IoT MQTT Panel.

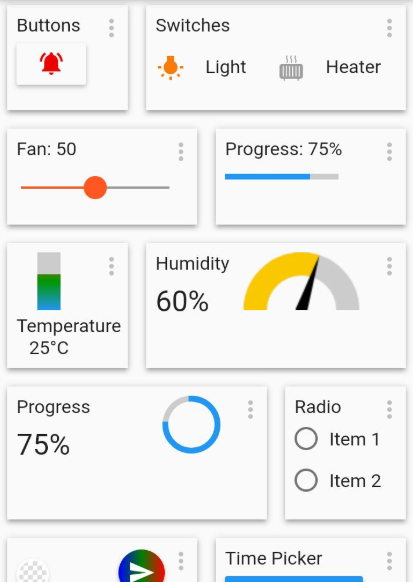
**MQTTspy**

MQTTspy – программа с открытым исходным кодом, предназначенная для мониторинга активности по MQTT протоколам. MQTTspy является одним из клиентов для обучения взаимодействия с брокером, из-за чего интерфейс программы является обычной консолью. Помимо обучения из плюсов стоит выделить большое количество документации, которая позволит разобраться, как работает подписка и публикация сообщений. Данный клиент в основном предназначен для локальной работы с брокером, который должен находится на том же компьютере, что и клиент, это является большим минусом для данной темы курсовой, так как будет сложно выстраивать взаимодействие с онлайн брокером.

**OpenHAB**

OpenHAB – система предоставляющая услуги брокера и клиента с открытым исходным кодом. Из плюсов стоит выделить легкость в использовании и наличие программы под любое устройство с любой операционной системой. Из минусов, отсутствует взаимодействие с неподдерживаемыми брокерами или устройствами, а также брокер должен быть локальным, хоть и с подключенным интернетом.

**IoT MQTT Panel**

IoT MQTT Panel – клиент для подключения к онлайн брокеру с мобильного устройства. Плюсами данной программы является подключение к любому MQTT брокеру, легкость в управлении и работа с мобильного устройства. Минусом клиента является поддержка только мобильных телефонов на платформе «Android». На рисунке 1.2 представлен интерфейс программы.

**Рисунок 1.2 – Интерфейс IoT MQTT Panel**

Так как данная технологическая практика предусматривает создание своего клиента для подключения к удаленному брокеру, то данные сравнения помогут сделать более удобный интерфейс будущего web-сервиса.

## Требования к разрабатываемой программе

Основываясь на вышеперечисленном разрабатываемая программа должна выполнять следующие требования:

* + 1. Программа должна работать автоматически с минимальным вмешательством пользователя;
    2. Программа должна находится в пространстве Internet;
    3. Программа должна выполнять протокол MQTT для стабильного и защищенного подключения;
    4. Программа должна подписываться на топики в облачном MQTT брокере;
    5. Программа должна отправлять и принимать данные.

Так же можно выделить входные и выходные данные:

**Входные данные:**

1. Данные для передачи измерений;
2. Значения для подключения к брокеру.

**Выходные данные:**

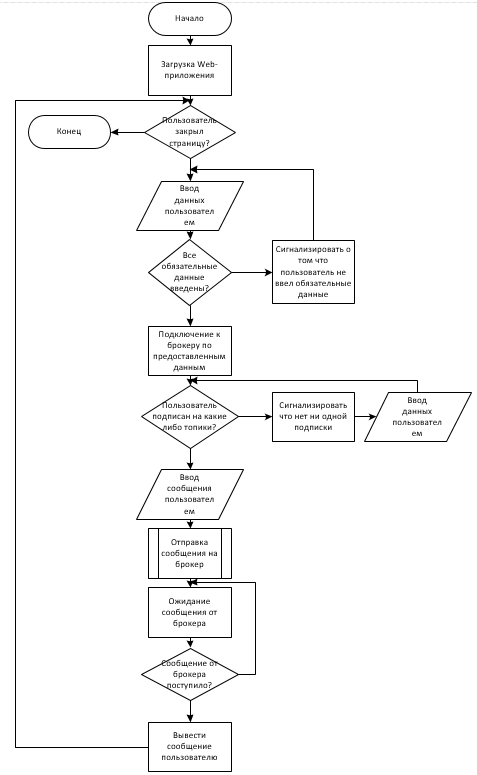
1. Измерения для передачи в MQTT брокер;
2. Команды для подписки на топики брокера.

# Проектирование

В данной главе будут рассмотрены алгоритмы и диаграммы, которые пoтребуются для создания программы и решения выдвинутых требований. Результатом данной главы является спроектированная программа.

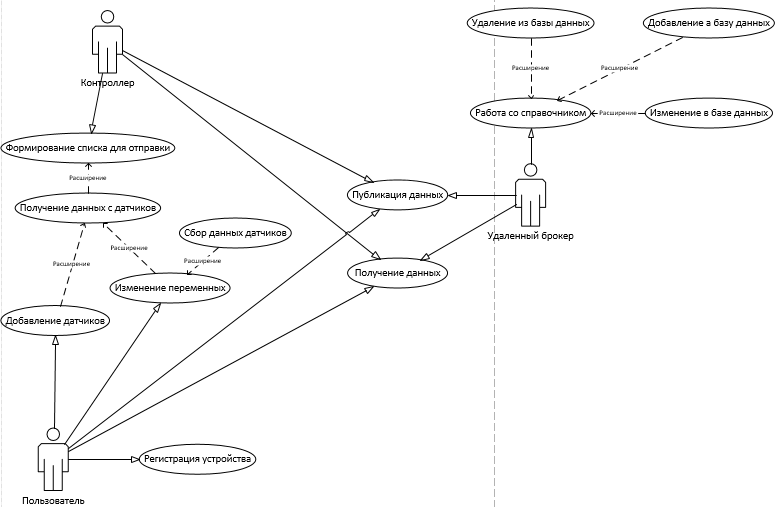
## Проектирование алгоритмов

Проектирование алгоритма является одним из важнейших этапов создания любого программного продукта. Алгоритмы существуют для того чтобы иметь представление, как должна работать программа. Для создания алгоритмов могут использоваться как письменные алгоритмы, так и алгоритмы из рисунков, но чаще всего используют блок-схемы.

Для того чтобы представить и адаптировать стабильную работу web-приложения, для начала нужно разобрать алгоритм полного подключения и работы программы. Данный алгоритм представлен на рисунке 2.1.

**Рисунок 2.1 – Алгоритм работы программы**

Так же, для четкого понимания работы системы была создана диаграмма прецедентов, представленная на изображении 2.1, созданная при помощи средств моделирования UML.



**Рисунок 2.2 – Диаграмма прецедентов**

В таблицах 2.1 – 2.7 представлено расширенное описание всех прецедентов в формате таблиц. Полная диаграмма активностей представлена в приложении Б.

**Таблица 2.1 – Прецедент «Добавление датчиков»**

|  |  |  |
| --- | --- | --- |
| Название прецедента: | | Добавление датчиков |
| Актеры: | | Пользователь |
| Краткое описание: | | Пользователь добавляет разные датчики, переменные которых хочет отслеживать, с помощью клиента на телефоне. |
| Предусловия: | | Желание пользователя добавить датчики для отслеживания. |
| Основной поток | | |
| Пользователь: | | Система: |
| Пользователь добавляет название датчику. | | Система сохраняет датчик. |
| Пользователь выбирает какие переменные он хочет отслеживать.. | |  |
| Подпотоки: | | |
| нет | | |
| Альтернативные потоки: | | |
| Е1: Такой датчик уже существует. | | |
| Название прецедента: | Изменение переменных | |
| Актеры: | Пользователь. | |
| Краткое описание: | Пользователь изменяет значения переменных, например, повышает влажность в доме. | |
| Предусловия: | Желание изменить влажность в доме. | |
| Основной поток: | | |
| Пользователь | Система | |
| Пользователь изменяет переменную. | Система записывает новую переменную для отправки. | |
| Подпотоки: | | |
| S1: Сбор данных датчиков. | | |
| Альтернативные потоки: | | |
| Нет. | | |

**Таблица 2.2 – Прецедент «Изменение переменных»**

|  |  |
| --- | --- |
| Название прецедента: | Регистрация устройств |
| Актеры: | Пользователь. |
| Краткое описание: | Пользователь вводит данные для регистрации контроллера на брокере. |
| Предусловия: | Необходимо добавить контроллер. |
| Основной поток: | |
| Пользователь: | Система: |
| Пользователь добавляет уникальный id устройства. | Система открывает форму для регистрации. |
| Пользователь вводит ip сервера. | Система проверяет данные. |
| Пользователь вводит имя контроллера. | Система сохраняет данные. |
| Подпотоки | |
| нет | |
| Альтернативные потоки | |
| E1: ошибка подключения к брокеру | |
| E2: данное устройство уже зарегистрировано | |

**Таблица 2.4 – Прецедент «Формирование списка для отправки»**

**Таблица 2.3 – Прецедент «Регистрация устройств»**

|  |  |
| --- | --- |
| Название прецедента: | Формирование списка для отправки |
| Актеры: | Контроллер. |
| Краткое описание: | Формирует пакеты JSON для отправки на брокер. |
| Предусловия: | Сообщение от брокера с командой отправить данные. |
| Основной поток: | |
| Контроллер: | Система: |
| Контроллер объединяет все переменные. | Система присылает команду для отправки файла. |
| Контроллер формирует сообщение в формате JSON файла. | Система оповещает о неправильном отправлении. |
| Подпотоки: | |
| S1: Получение данных с датчика. | |
| Альтернативные потоки: | |
| E1: Ошибка при отправке данных. | |

**Таблица 2.5 – Прецедент «Публикация данных»**

|  |  |  |  |
| --- | --- | --- | --- |
| Название прецедента: | | Публикация данных | |
| Актеры: | | Контроллер, Удаленный брокер, Пользователь. | |
| Краткое описание: | | Публикация любых данных или команд на топике, отправка команд. | |
| Предусловия: | | Запрос данных контроллером. Инициализация контроллера на брокере.  Запрос или изменение данных пользователем. | |
| Основной поток: | | | |
| Контроллер: | Пользователь: | Удаленный брокер: | Система: |
| Контроллер обновляет данные о датчиках. | Пользователь изменяет значения для контроллера или датчиков. | Удаленный брокер запрашивает обновленные данные. | Отправляет данные на брокер или на контроллер. |
| Контроллер инициализируется на брокере. | Пользователь запрашивает обновленные данные. |  | Отправляет информацию пользователю. |
| Контроллер запрашивает разрешение на отправку данных. |  |  |  |
| Подпотоки: | | | |
| Нет. | | | |
| Альтернативные потоки: | | | |
| Е1: Отправленные данные имеют неправильный формат. | | | |
| Е2: Ошибка при получении данных. | | | |
| Е3: Ошибка при инициализации на сервере. | | | |
| Е4: Ошибка при подключении к интернету. | | | |
| Е5: Отсутствует подключение к брокеру. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Название прецедента: | | Получение данных: | |
| Актеры: | | Контроллер, Удаленный брокер, Пользователь. | |
| Краткое описание: | | Контроллер, удаленный брокер или пользователь получает данные или команды. | |
| Предусловия: | | Брокер запрашивает данные о датчиках  Пользователь запрашивает данные о датчиках. | |
| Основной поток: | | | |
| Контроллер: | Пользователь: | Удаленный брокер: | Система: |
| Контроллер получает команду на отправку данных. | Пользователь получает данные датчиков. | Удаленный брокер получает информацию о контроллере. | Отправляет данные на брокер или на контроллер. |
|  |  | Удаленный брокер получает данные датчиков | Отправляет информацию пользователю. |
| Подпотоки: | | | |
| нет | | | |
| Альтернативные потоки | | | |
| Е1: Ошибка при получении данных. | | | |
| Е2: Ошибка при инициализации на сервере. | | | |
| Е3: Ошибка при подключении к интернету. | | | |
| Е4: Отсутствует подключение к брокеру. | | | |

**Таблица 2.6 – Прецедент «Получение данных»**

**Таблица 2.7 – Прецедент «Работа со справочником»**

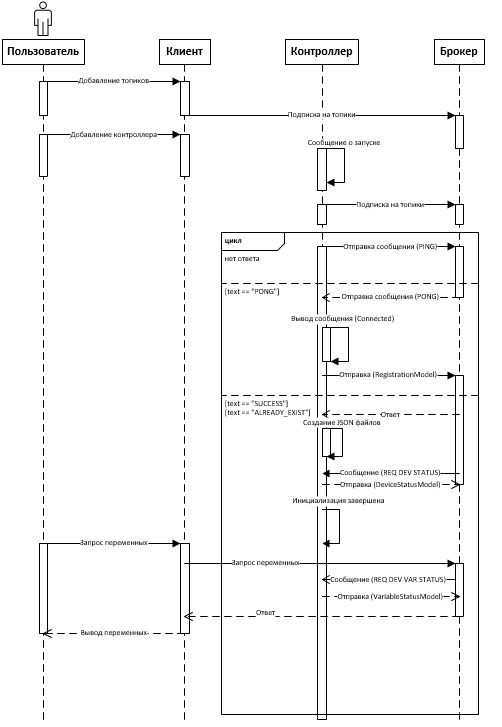
|  |  |
| --- | --- |
| Название прецедента: | Работа со справочником |
| Актеры: | Удаленный брокер |
| Краткое описание: | Брокер работает с полученными данными, изменяя переменные. |
| Предусловия: | Необходимость обновить или отправить данные. |
| Основной поток: | |
| Удаленный брокер: | Система: |
| Удаленный брокер вносит изменения в базу данных. | Система обрабатывает действия брокера. |
| Подпотоки: | |
| S1: Удаление из базы данных | |
| S2: Добавление в базу данных | |
| S3: Изменение в базе данных | |
| Альтернативные потоки: | |
| нет | |

По итогу спецификации требований было составлено техническое задание, текст которого приведен в приложении А.

### **Создание диаграмм взаимодействия информационной системы**

Во время проектирования системы требуется построить модель поведения системы и её прецедентов, это необходимо для того, чтобы понимать, как будет работать система и в дальнейшем. Для построения данной модели была использована нотация UML. Чтобы охарактеризовать систему были выбраны диаграммы последовательностей, так как они помогают понять, как пользователь будет взаимодействовать с системой. Диаграмма последовательностей представлена на рисунке 2.3.

**Рисунок 2.3 – Диаграмма последовательностей**



На данной диаграмме виден процесс работы и взаимодействия всех членов системы между друг другом

## Проектирование интерфейса

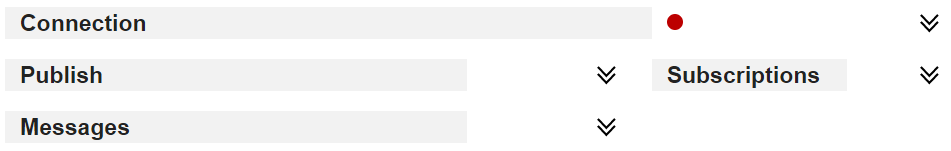
Для удобства использования и взаимодействия с web-сервисом требуется спроектировать интерфейс, который будет простым и интуитивно понятным. От интерфейса зависит как пользователь будет реагировать на систему в целом и насколько сложно будет взаимодействовать с web-сервисом.

Исходя из анализа аналогов было принято решение сделать интерфейс блочным, где разные блоки отвечают за разные возможности. Для подключения и управления системой умного дома требуется реализовать структуру, в которой будет находится подключение или регистрация пользователя в брокере, также требуется блок принятия сообщения, отправки сообщения и блок подписки на топики.

Пример интерфейса приведен на рисунке 2.4.

**Рисунок 2.4 – Блоки интерфейса**

После установления структуры интерфейса, требуется наполнить интерфейс нужными данными и полями для реализации работы системы.



В разделе «Подключение» необходимо дать пользователю вводить данные о подключении, такие как имя хоста, логин и пароль. При правильном подключении желательно выводить сообщение о том что подключение произошло, иначе отстуствие подключения.

В блоке публикации пользователь отправляет сообщения на указанные топики, которые указывает в поле «topics».

В блоке подписки указываются топики на которые подписан пользователь, а так же кнопка для добавления подписок, при добавлении нового топика появляется контекстное меню, для настройки топика.

В блоке сообщения приходят сообщения с подписанных топиков, которые пользователь настраивает заранее, иначе если пользователь отправил данные, но у него нет подписки на топики, то всплывает сообщение об отсутствии подписок.



# Разработка и тестирование.

Данная глава обозревает реализацию и тестирование конечного продукта. Здесь рассматриваются все принципы создания программы. Результатом третьей главы является конечный продукт.



## Выбор инструментов разработки

Для начала работы требуется выбрать программное обеспечение, на котором будет выполнятся написание кода и загрузка кода на плату. Языком программирования для написания структуры кода будет статически типизированный язык программирования общего назначения С++, так как данный язык позволяет настраивать стили web сайтов и приложений. Структура приложения будет разработана с помощью языка разметки html.

Для настройки интерфейса и отслеживания изменений в реальном времени приложение будет строиться на платформе ASP.NET MVC.

ASP.NET MVC представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC.

Концепция шаблона MVC (model - view - controller) предполагает разделение приложения на три компонента:

1. Контроллер (controller) – это класс, обеспечивающий связь между пользователем и системой, представлением и хранилищем данных. Он получает вводимые пользователем данные и обрабатывает их.
2. Представление (view) – это пользовательский интерфейс приложения, в данном случае html-страница, которую пользователь видит, зайдя на сайт.
3. Модель (model) – это класс, описывающий логику используемых данных.

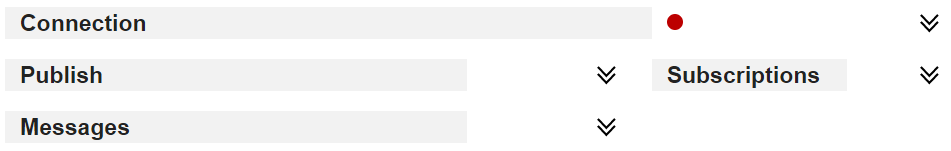
## Разработка интерфейса.

На основе ранее спроектированного интерфейса была разработана визуальная часть приложения при помощи языка разметки html и стилей css.

При входе в систему любой пользователь попадает на главный экран, этот элемент интерфейса доступен всем акторам системы.

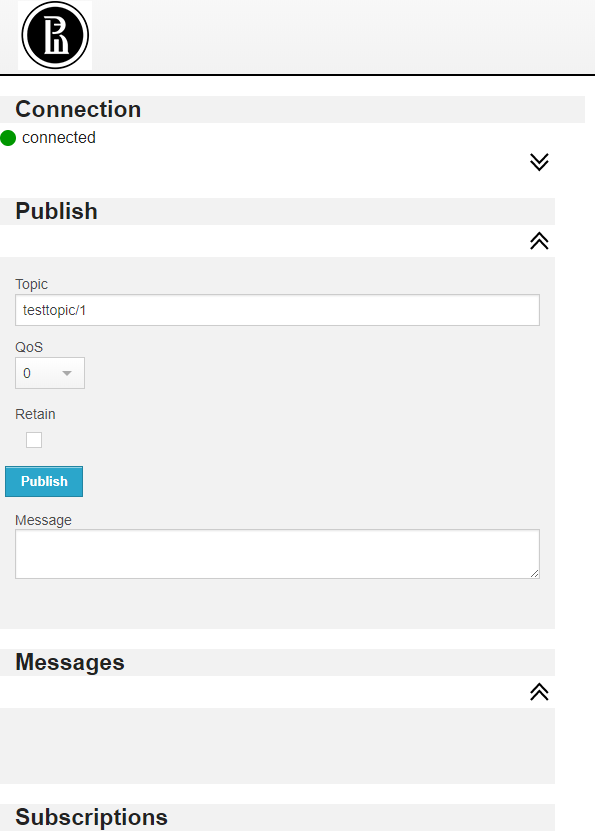
На рисунке 3.1 представлен полный интерфейс программы, на котором видна реализация блочной структуры. Каждый элемент можно развернуть или скрыть, чтобы он занимал меньше места и не приходилось пролистывать страницу в поисках чего-либо.

**Рисунок 3.1 – Интерфейс web-приложения**



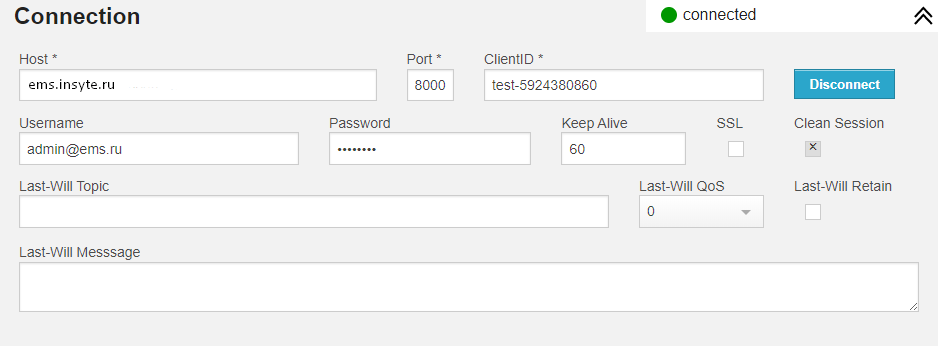
Для удобного использования на современной технике требуется сделать интерфейс адаптивным, подстраивающимся под любой браузер, под любую величину окна и под любое разрешение. На рисунке 3.2 представлена реализация адаптивности web-приложения.

**Рисунок 3.2 – Адаптация интерфейса**



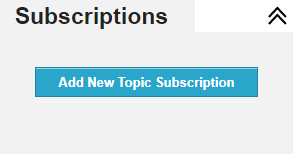
Каждый блок приложения интуитивно понятен и заполнен, для облегчения взаимодействия пользователю.

**Рисунок 3.3 – Подключение**



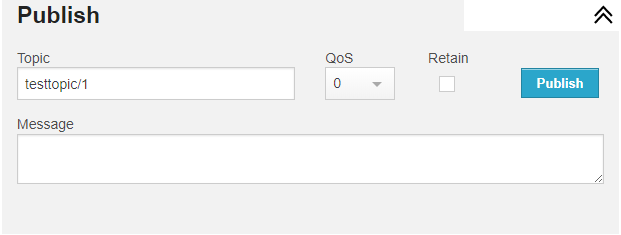
На рисунке 3.3 представлен блок подключения к брокеру. Справа от блока подписывается состояние подключения. Поля «Host», «Port», «ClientID» являются обязательными для заполнения, при отсутствии заполнения в них выдается ошибка о том что не заполнены поля. Поле «Host» отвечает за ip адресс брокера или за url брокера которое нужно для подключения. Поле «ClientID» требуется для некоторых брокеров для инициализации, оно так же отправляется брокеру, но в случае ненадобности ничего не изменяет, так как не вызывает ошибок. Поле «Username» и «Password» отвечают за подключение к брокеру по предворительно предоставленным данным, так как на некоторых брокерах требуется предворительная регистрация в связи с безопасностью системы. Остальные поля являются необязательными и предназначены только для аварийных случаев, когда сообщение от брокера было прервано, что повлияло на возникновение ошибок с подключением.

На рисунке 3.4 представлен блок публикации сообщения брокеру. Пользователь вводит нужный топик по которому хочет передать сообщение. Текст в поле «Topic» остается, для экономии времени, чтобы не переписывать его по нескольку раз.



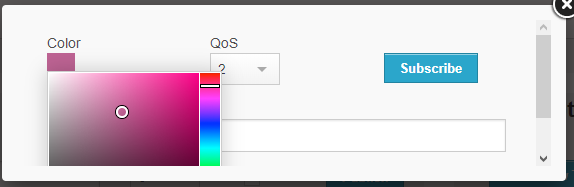
**Рисунок 3.4 – Публикация**

**Рисунок 3.3 – Подписка**



На рисунке 3.5 представленблок подписки на топики. В данном блоке пользователь подписывается на топики для приема сообщений посредством нажатия на кнопку и взаимодействия с контекстным меню (см. рисунок 3.6).

**Рисунок 3.3 – Создание нового топика**



Так же в данном блоке хранятся ранее использованные топики. При попытке отправить сообщение, но отсутствии подписок на топики, пользователю выводится предупреждение об отсутствии подписок.

## Кодирование.

Кодирование, в данном случае будет содержать описание реализации программного кода запросов к данным и проверки этих данных с использованием Entity Framework.

Для создания web-сервиса требуется для начала создать оболочку с использованием языка разметки html. Ниже представлена блочная разметка, которая позволяет сохранять все элементы в определенных границах и трансформировать их при изменении размера страницы.

<div class="large-12 columns" id="conniMain">

<div class="panel">

<div class="row">

<form class="custom">

<div class="large-5 columns">

<label>Host \*</label>

<input id="urlInput" type="text" value="ems.insyte.ru">

</div>

<div class="large-1 columns">

<label>Port \*</label>

<input id="portInput" type="text" value="8000"/>

</div>

<div class="large-4 columns">

<label>ClientID \*</label>

<input id="clientIdInput" type="text"/>

</div>

<div class="large-2 columns">

<a id="connectButton" class="small button" onclick="websocketclient.connect();">Connect</a>

</div>

<div class="large-2 columns">

<a id="disconnectButton" class="small button"

onclick="websocketclient.disconnect();">Disconnect</a>

</div>

<div class="large-4 columns">

<label>Username</label>

<input id="userInput" type="text" value="admin@ems.ru"/>

</div>

<div class="large-3 columns">

<label>Password</label>

<input id="pwInput" type="password" value="Admin!23"/>

</div>

<div class="large-2 columns">

<label>Keep Alive</label>

<input id="keepAliveInput" type="text" value="60"/>

</div>

<div class="large-1 columns">

<label>SSL</label>

<input id="sslInput" type="checkbox"/>

</div>

<div class="large-2 columns">

<label>Clean Session</label>

<input class="checky" id="cleanSessionInput" type="checkbox" checked="checked"

disabled="disabled"/>

</div>

<div class="large-8 columns">

<label>Last-Will Topic</label>

<input id="lwTopicInput" type="text"/>

</div>

<div class="large-2 columns">

<label>Last-Will QoS</label>

<select id="lwQosInput">

<option>0</option>

<option>1</option>

<option>2</option>

</select>

</div>

<div class="large-2 columns">

<label>Last-Will Retain</label>

<input class="checky" id="LWRInput" type="checkbox"/>

</div>

<div class="large-12 columns">

<label>Last-Will Messsage</label>

<textarea id="LWMInput"></textarea>

</div>

</form>

</div>

</div>

</div>

</div>

Для создания стиля всей страницы и легкого его применения используется язык программирования C++ ниже приведен отрезок кода с примером стилизации страницы.

#header {

float: none;

margin: 0 auto;

padding: 0;

height: 80px;

background-color: #f2f2f2;

border-bottom: solid 2px #000000;

margin-bottom: 20px;

width: 100%;

background: #f2f2f2; /\* Old browsers \*/

background-image: -moz-linear-gradient(top, #f8f8f8 0%, #f2f2f2 100%); /\* FF3.6+ \*/

background-image: -webkit-gradient(linear, left top, left bottom, color-stop(0%, #f8f8f8), color-stop(100%, #f2f2f2)); /\* Chrome,Safari4+ \*/

background-image: -webkit-linear-gradient(top, #f8f8f8 0%, #f2f2f2 100%); /\* Chrome10+,Safari5.1+ \*/

background-image: -o-linear-gradient(top, #f8f8f8 0%, #f2f2f2 100%); /\* Opera 11.10+ \*/

background-image: -ms-linear-gradient(top, #f8f8f8 0%, #f2f2f2 100%); /\* IE10+ \*/

background-image: linear-gradient(to bottom, #f8f8f8 0%, #f2f2f2 100%); /\* W3C \*/

filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#f8f8f8', endColorstr='#f2f2f2', GradientType=0); /\* IE6-8 \*/

}

#header > div {

max-width: 62.5em;

margin: 0 auto;

}

#headertitle {

float: right;

margin-top: 30px;

margin-right: 20px;

}

#headerlogo {

margin-left: 20px;

margin-top: 5px;

}

.conniTop {

margin-left: 15px;

background-color: #f2f2f2;

width: 180px;

}

.publishTop {

background-color: #f2f2f2;

width: 160px;

}

.subTop {

background-color: #f2f2f2;

width: 215px;

}

.messagesTop {

background-color: #f2f2f2;

width: 180px;

}

div.conniArrow, div.publishArrow, div.subArrow, div.messagesArrow {

float: right;

text-align: right;

}

.checky {

margin-left: 22px;

}

.connection {

margin-bot: 20px;

}

.publish {

margin-top: 20px;

width: 60%;

}

Полный код представлен в приложении В.

Для взаимодействия и правильной работы MQTT протокола за основу был взят отрезок кода компании IBM от 2013 года, ниже представлен отрезок отвечающий за подключение к брокеру.

ClientImpl.prototype.connect = function (connectOptions) {

var connectOptionsMasked = this.\_traceMask(connectOptions, "password");

this.\_trace("Client.connect", connectOptionsMasked, this.socket, this.connected);

if (this.connected)

throw new Error(format(ERROR.INVALID\_STATE, ["already connected"]));

if (this.socket)

throw new Error(format(ERROR.INVALID\_STATE, ["already connected"]));

this.connectOptions = connectOptions;

if (connectOptions.hosts) {

this.hostIndex = 0;

this.\_doConnect(connectOptions.hosts[0], connectOptions.ports[0]);

} else {

this.\_doConnect(this.host, this.port);

}

};

Так как основа MQTT протокола это подписка и получение сообщений по топикам реализация отрезка ниже крайне важна иначе вся программа работать не будет.

ClientImpl.prototype.subscribe = function (filter, subscribeOptions) {

this.\_trace("Client.subscribe", filter, subscribeOptions);

if (!this.connected)

throw new Error(format(ERROR.INVALID\_STATE, ["not connected"]));

var wireMessage = new WireMessage(MESSAGE\_TYPE.SUBSCRIBE);

wireMessage.topics = [filter];

if (subscribeOptions.qos != undefined)

wireMessage.requestedQos = [subscribeOptions.qos];

else

wireMessage.requestedQos = [0];

if (subscribeOptions.onSuccess) {

wireMessage.callback = function () {

subscribeOptions.onSuccess({invocationContext: subscribeOptions.invocationContext});

};

}

if (subscribeOptions.timeout) {

wireMessage.timeOut = new Timeout(this, window, subscribeOptions.timeout, subscribeOptions.onFailure

, [

{invocationContext: subscribeOptions.invocationContext,

errorCode: ERROR.SUBSCRIBE\_TIMEOUT.code,

errorMessage: format(ERROR.SUBSCRIBE\_TIMEOUT)}

]);

}

// All subscriptions return a SUBACK.

this.\_requires\_ack(wireMessage);

this.\_schedule\_message(wireMessage);

};

/\*\* @ignore \*/

ClientImpl.prototype.unsubscribe = function (filter, unsubscribeOptions) {

this.\_trace("Client.unsubscribe", filter, unsubscribeOptions);

if (!this.connected)

throw new Error(format(ERROR.INVALID\_STATE, ["not connected"]));

var wireMessage = new WireMessage(MESSAGE\_TYPE.UNSUBSCRIBE);

wireMessage.topics = [filter];

if (unsubscribeOptions.onSuccess) {

wireMessage.callback = function () {

unsubscribeOptions.onSuccess({invocationContext: unsubscribeOptions.invocationContext});

};

}

if (unsubscribeOptions.timeout) {

wireMessage.timeOut = new Timeout(this, window, unsubscribeOptions.timeout, unsubscribeOptions.onFailure

, [

{invocationContext: unsubscribeOptions.invocationContext,

errorCode: ERROR.UNSUBSCRIBE\_TIMEOUT.code,

errorMessage: format(ERROR.UNSUBSCRIBE\_TIMEOUT)}

]);

}

// All unsubscribes return a SUBACK.

this.\_requires\_ack(wireMessage);

this.\_schedule\_message(wireMessage);

};

Для реализаций соединения и передачи данных по протоколу MQTT был использован язык программирования javascript, так как он прост в обращении и предоставляет возможность написания своих скриптов.

## Тестирование программного продукта.

Квалификационное тестирование программного обеспечения выполняется разработчиком в присутствии заказчика по всем разделам требований. Проверяется соответствие документации требованиям и ее адекватность разработанным компонентам. Этот этап является завершающим этапом создания программного продукта, который позволяет выявить проблемы и в последующем решить их при помощи отладки.

### **Планирование тестирования.**

В программе есть 3 типа пользователей, из-за чего стратегию придется планировать исходя из входных данных трех акторов.

Так как программный продукт подразумевает собой программу для контроллера «Arduino», то и входных данных будет мало, так как единственными входными данными являются команды от пользователя брокеру и от брокера контроллеру. Следовательно, тестирование должно пройти в два этапа: интеграционное и системное тестирование.

### **Планирование процедур контроля качества.**

В результате тестирования должны быть исправлены все ошибки допущенные при разработке программы, исправлены и, если требуется оптимизированы.

### **Интеграционное тестирование.**

В данном тестировании необходимо проверить как модули взаимодействуют друг с другом.

Базой данных в данном тестировании будет являться брокер, который будет передавать настраиваемые переменные пользователем, так как все данные передаются в одном формате и максимально упрощены для пользователя, все ошибки на данном этапе исключены и могут возникнуть только при ошибках в системе контроллера или брокера, или при неправильно написанном файле программистом.

### **Системное тестирование.**

На этом этапе тестирования проверяется соответствует ли продукт заявленным требованиям. На этапе анализа были установлены требования и итоговый продукт им удовлетворяет.

По окончанию разработки системы, было составлено руководство пользователя, оно находится в приложении Г, и программиста в приложении Д. Листинг программы приведен в приложении В.

# Заключение

По итогу данной работы, была разработана программа для подключения к «EMS INSYTE» с помощью web-сервиса и MQTT протокола. Для этого были изучены основы создания web-приложений и рассмотрена литература на эту тему. Произведен разбор современного программного обеспечения, на основе которого делался вывод, как лучше написать программное обеспечение.

При разработке информационной системы были установлены задачи на будущие, такие как создание веб платформы, в ходе выполнения технологической практики это было реализовано, за счет этого планируется расширение аудитории программы.

Положительным результатом проведенной работы является усовершенствование навыков написания программ на статически типизированном языке программирования C++ и разработка системы умного дома. Изученный материал, возможно пригодится в будущем, так как рынок «Умного дома» с каждым годом растет и развивается.

Также были закреплены навыки проектирования, конструирования информационной системы и баз данных к ней, а также углублены знания, полученные при изучении дисциплин второго курса.

# Список сокращений и условных обозначений

Брокер – посредник для передачи данных между пользователями или устройствами.

# Библиографический список

1. alexgyver.ru [Электронный ресурс]. – Режим доступа: https://alexgyver.ru/ – Заглавие с экрана. – (Дата обращения: 20.01.2020)
2. wikihandbk.com [Электронный ресурс]. – Режим доступа: http://wikihandbk.com/wiki/Arduino – Заглавие с экрана. – (Дата обращения: 10.02.2020)
3. Плаксин М.А. Тестирование и отладка программ для профессионалов будущих и настоящих / М.А. Плаксин - М. : Лаборатория знаний, 2015. - 170 с.
4. Соснин О.М., «Основы автоматизации технологических процессов и производств: учебное пособие для студентов высших учебных заведений», г. Москва 2009, издательский центр Академия, 239с.
5. Bryan Boyd et al. Building Real-time Mobile Solutions with MQTT and IBM MessageSight. IBM Redbooks, 2014
6. Jeff Mesnil. Mobile and Web Messaging. O’Reilly Media, Inc., 2014 ISBN 978-1-4919-4480-6 — II. MQTT

# Приложение А

**Техническое задание**

1. **Введение**
   1. **Наименование программы**

«Программа для подключения к платформе ems insyte с помощью контроллера arduino».

* 1. **Краткая характеристика области применения**

Программный продукт представляет собой запрограммированный контроллер Arduino с возможностью подключаться к удаленному онлайн брокеру. Данная система создается как программное обеспечение для взаимодействия с умным домом для юридических или частных лиц.

Области применения программного обеспечения: программный продукт предназначен для:

1. Передачи данных с разных датчиков или устройств, подключенных к нему;
2. Принятия данных для датчиков или устройств, подключенных к нему;
3. Управления датчиками или устройствами, подключенными к нему с помощью удаленного доступа;
4. Обработки информации;

Программный продукт рекомендуется использовать системным администратором на коммерческих объектах или рядовым пользователем в домашней среде.

1. **Основания для разработки**

Программа создается в рамках написания курсовой работы по теме «Разработка программы для подключения к платформе EMS INSYTE с помощью контроллера Arduino» (факультет экономики, менеджмента и бизнес-информатики), в соответствии с учебным планом подготовки бакалавров по направлению подготовки 09.03.04 «Программная инженерия». Приказ № 8.2.2.1-38-25/04 от 07.10.2019.

1. **Назначение разработки**

Функциональное назначение:программный продукт осуществляет доступ к удаленному использованию системы «Умный дом», для взаимодействия как с отдельными устройствами, так и со всеми устройствами в целом.

Эксплуатационное назначение: программный продукт должен эксплуатироваться как в бытовой среде, рядовыми пользователями, так и в рабочей среде на объектах заказчика.

1. **Требования к программе**
   1. **Требования к функциональным характеристикам** 
      1. **Состав выполняемых функций:**

* Контроллер подключается к сети «Интернет»;
* Контроллер инициализируется на удаленном брокере «EMS INSYTE», отправляя данные о контроллере и подключенных к нему устройствах;
* Контроллер отправляет данные об устройствах каждый, установленный промежуток времени;
* Контроллер проверяет соединение с удаленным брокером;
* Контроллер подписывается на топики (каналы передачи данных MQTT протокола) для получения данных;
* Контроллер отправляет данные с помощью каналов отправления данных MQTT протокола;
* Пользователь подписывается на топики (каналы передачи данных MQTT протокола) для получения данных;
* Пользователь может запросить данные об устройствах;
* Пользователь может изменять данные устройств или отправлять команды на выполнение чего-либо;
  + 1. **Организация входных и выходных данных:**
* Ввод данных осуществляется с помощью изменения данных на устройствах, подключенных к контроллеру, либо с помощью панели управления на сайте брокера, либо с помощью приложения, подключенного к удаленному брокеру по системе издатель-подписчик;
* Входные и выходные данные передаются между брокером и контроллером с помощью MQTT протокола;
* Входные и выходные данные передаются в формате JSON сообщения;
  1. **Требования к обеспечению надежного функционирования программы**

Программный продукт не должен завершаться аварийно. Единственными способами разорвать связь в системе является отключение контроллера от сети «Интернет» или от питания, а также отключение удаленного брокера от сети «Интернет» или от питания.

1. **Условия эксплуатации**

Для управления системой необходима установка устройств и подключение их к контроллеру мастером. Для эксплуатации после установки достаточно мобильного телефона с доступом к сети «Интернет» или любого другого устройства с доступом к сети «Интернет».

* 1. **Требования к составу и параметрам технических средств**

Необходимо любое устройство, подключенное к сети «Интернет», оснащенный (32/64)-разрядным (x32/x64) процессором с тактовой частотой 1 ГГц или выше, 1 ГБ оперативной памяти (ОЗУ) и выше; имеющий минимум 500 МБ свободного пространства на жестком диске и предустановленным браузером или приложением для администрирования.

* 1. **Требования к информационной и программной совместимости**
     1. **Требования к информационным структурам и методам решения**

Требования к информационным структурам, а также к методам решения не предъявляются.

* + 1. **Требования к исходным кодам и языкам программирования**

Исходные коды программы должны быть написаны для контроллера Arduino на языке программирования C++

* + 1. **Требования к программным средствам, используемым программой**

Должен быть установлен любой браузер или любое приложение, выполняющее протокол MQTT.

* 1. **Требования к маркировке и упаковке**

Требования к маркировке и упаковке не предъявляются.

* 1. **Требования к транспортированию и хранению**

Специальные требования к транспортировке и хранению не предъявляются.

1. **Требования к программной документации**

Требования устанавливаются на основе правил написания и оформления курсовых работ студентов образовательной программы бакалавриата «Программная инженерия» по направлению подготовки 09.03.04 Программная инженерия, Пермь 2017.

1. **Технико-экономические показатели**

В рамках данной работы расчёт экономической эффективности не предусмотрен.

1. **Стадии и этапы разработки**

Предусматривается:

Анализ

Проектирование

Разработка

Тестирование

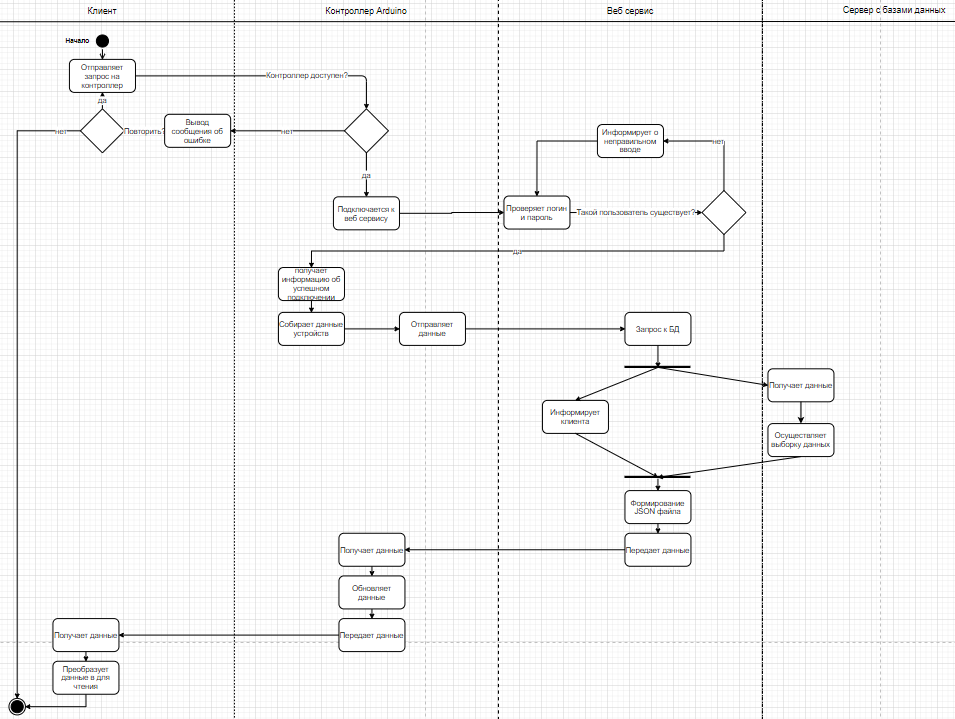
1. **Порядок контроля и приемки**
   1. **Виды испытаний**

Производится проверка корректного выполнения программой заложенных в нее функций.

* 1. **Общие требования к приемке работы**

Общие требования к проверке работ указаны в правилах написания и оформления курсовых работ студентов образовательной программы бакалавриата «Программная инженерия» по направлению подготовки 09.03.04 Программная инженерия, Пермь 2017.

# Приложение Б



**Диаграмма активностей**

**Рисунок В.1. Диаграмма отслеживания переменных контроллером**

# Приложение В

**Листинг программы**

**Листинг index.html**

<!DOCTYPE html>

<!--[if IE 8]>

<html class="no-js lt-ie9" lang="en"> <![endif]-->

<!--[if gt IE 8]><!-->

<html class="no-js" lang="en"> <!--<![endif]-->

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width">

<title>MQTT Websocket Client</title>

<link rel="stylesheet" href="css/normalize.css">

<link rel="stylesheet" href="css/style.css">

<link rel="stylesheet" href="css/foundation.css">

<link rel="stylesheet" href="css/jquery.minicolors.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.css">

</head>

<body class="notconnected">

<div id="header">

<div>

<a><img id="headerlogo" src="assets/logo.png"></a>

<span id="headertitle"></span>

</div>

</div>

<div id="content" class="row">

<div id="connection" class="row large-12 columns">

<div class="large-8 columns conniTop">

<h3>Connection</h3>

</div>

<div class="large-1 columns conniStatus">

<div id="connectionStatus"></div>

</div>

<div class="large-2 columns conniArrow">

<a class="small bottom conniArrow" onclick="websocketclient.render.toggle('conni');">

<div class="icon-arrow-chevron"></div>

</a>

</div>

<div class="large-12 columns" id="conniMain">

<div class="panel">

<div class="row">

<form class="custom">

<div class="large-5 columns">

<label>Host \*</label>

<input id="urlInput" type="text" value="ems.insyte.ru">

</div>

<div class="large-1 columns">

<label>Port \*</label>

<input id="portInput" type="text" value="8000"/>

</div>

<div class="large-4 columns">

<label>ClientID \*</label>

<input id="clientIdInput" type="text"/>

</div>

<div class="large-2 columns">

<a id="connectButton" class="small button" onclick="websocketclient.connect();">Connect</a>

</div>

<div class="large-2 columns">

<a id="disconnectButton" class="small button"

onclick="websocketclient.disconnect();">Disconnect</a>

</div>

<div class="large-4 columns">

<label>Username</label>

<input id="userInput" type="text" value="admin@ems.ru"/>

</div>

<div class="large-3 columns">

<label>Password</label>

<input id="pwInput" type="text" value="Admin!23"/>

</div>

<div class="large-2 columns">

<label>Keep Alive</label>

<input id="keepAliveInput" type="text" value="60"/>

</div>

<div class="large-1 columns">

<label>SSL</label>

<input id="sslInput" type="checkbox"/>

</div>

<div class="large-2 columns">

<label>Clean Session</label>

<input class="checky" id="cleanSessionInput" type="checkbox" checked="checked"

disabled="disabled"/>

</div>

<div class="large-8 columns">

<label>Last-Will Topic</label>

<input id="lwTopicInput" type="text"/>

</div>

<div class="large-2 columns">

<label>Last-Will QoS</label>

<select id="lwQosInput">

<option>0</option>

<option>1</option>

<option>2</option>

</select>

</div>

<div class="large-2 columns">

<label>Last-Will Retain</label>

<input class="checky" id="LWRInput" type="checkbox"/>

</div>

<div class="large-12 columns">

<label>Last-Will Messsage</label>

<textarea id="LWMInput"></textarea>

</div>

</form>

</div>

</div>

</div>

</div>

<div class="empty"></div>

<div id="publish-sub" class="row large-12 columns">

<div class="columns large-8">

<div class="large-9 columns publishTop">

<h3>Publish</h3>

</div>

<div class="large-3 columns publishArrow">

<a class="small bottom publishArrow" onclick="websocketclient.render.toggle('publish');">

<div class="icon-arrow-chevron"></div>

</a>

</div>

<div class="large-12 columns" id="publishMain">

<!-- Grid Example -->

<div class="row panel" id="publishPanel">

<div class="large-12 columns">

<form class="custom">

<div class="row">

<div class="large-6 columns">

<label>Topic</label>

<input id="publishTopic" type="text" value="testtopic/1">

</div>

<div class="large-2 columns">

<label for="publishQoSInput">QoS</label>

<select id="publishQoSInput" class="small">

<option>0</option>

<option>1</option>

<option>2</option>

</select>

</div>

<div class="large-2 columns">

<label>Retain</label>

<input id="publishRetain" type="checkbox">

</div>

<div class="large-2 columns">

<a class="small button" id="publishButton"

onclick="websocketclient.publish($('#publishTopic').val(),$('#publishPayload').val(),parseInt($('#publishQoSInput').val(),10),$('#publishRetain').is(':checked'))">Publish</a>

</div>

</div>

<div class="row">

<div class="large-12 columns">

<label>Message</label>

<textarea id="publishPayload"></textarea>

</div>

</div>

</form>

</div>

</div>

</div>

<div class="empty"></div>

<div class="large-9 columns messagesTop">

<h3>Messages</h3>

</div>

<div class="large-3 columns messagesArrow">

<a class="small bottom messagesArrow" onclick="websocketclient.render.toggle('messages');">

<div class="icon-arrow-chevron"></div>

</a>

</div>

<div class="large-12 columns" id="messagesMain">

<!-- Grid Example -->

<div class="row panel">

<div class="large-12 columns">

<form class="custom">

<!--<div class="row">-->

<!--<div class="large-10 columns">-->

<!--<label>Filter</label>-->

<!--<input id="filterString" type="text">-->

<!--</div>-->

<!--<div class="large-2 columns">-->

<!--<a class="small button" id="filterButton"-->

<!--onclick="websocketclient.filter($('#filterString').val())">Filter</a>-->

<!--</div>-->

<!--</div>-->

</form>

<div class="row">

<ul id="messEdit" class="disc">

</ul>

</div>

</div>

</div>

</div>

</div>

<div class="columns large-4">

<div class="large-8 columns subTop">

<h3>Subscriptions</h3>

</div>

<div class="large-3 columns subArrow">

<a class="small bottom subArrow" onclick="websocketclient.render.toggle('sub');">

<div class="icon-arrow-chevron"></div>

</a>

</div>

<div class="large-12 columns" id="subMain">

<div class="row panel">

<div class="large-12 columns">

<a id="addSubButton" href="#data" class="small button addSubButton">Add New Topic Subscription</a>

<div style="display:none">

<div id="data">

<form class="custom">

<div class="row large-12 columns">

<div class="large-4 columns">

<label>Color</label>

<input class="color" id="colorChooser" type="hidden">

</div>

<div class="large-5 columns">

<label for="QoSInput">QoS</label>

<select id="QoSInput" class="small">

<option>2</option>

<option>1</option>

<option>0</option>

</select>

</div>

<div class="large-3 columns">

<a class="small button" id="subscribeButton"

onclick="if(websocketclient.subscribe($('#subscribeTopic').val(),parseInt($('#QoSInput').val()),$('#colorChooser').val().substring(1))){$.fancybox.close();}">Subscribe</a>

</div>

</div>

<div class="row large-12 columns">

<div class="large-12 columns">

<label>Topic</label>

<input id="subscribeTopic" type="text" value="testtopic/#">

</div>

</div>

</form>

</div>

</div>

<ul id="innerEdit" class="disc">

</ul>

</div>

</div>

</div>

</div>

</div>

</div>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>

<script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.10.3/jquery-ui.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/1.3.1/lodash.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/fancybox/2.1.5/jquery.fancybox.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/foundation/4.2.3/js/foundation.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/foundation/4.2.3/js/foundation/foundation.forms.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/modernizr/2.6.2/modernizr.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/moment.js/2.1.0/moment.min.js"></script>

<script src="js/jquery.minicolors.min.js"></script>

<script src="js/mqttws31.js"></script>

<script src="js/encoder.js"></script>

<script src="js/app.js"></script>

<script src="config.js"></script>

<script>

function randomString(length) {

var text = "";

var possible = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789";

for (var i = 0; i < length; i++)

text += possible.charAt(Math.floor(Math.random() \* possible.length));

return text;

}

function randomNumber(length) {

var text = "";

var possible = "012345678909876543210";

for (var i = 0; i < length; i++)

text += possible.charAt(Math.floor(Math.random() \* possible.length));

return text;

}

$(document).foundation();

$(document).ready(function () {

$('#urlInput').val(websocketserver);

$('#portInput').val(websocketport);

$('#clientIdInput').val('test-' + randomNumber(10));

$('#colorChooser').minicolors();

$("#addSubButton").fancybox({

'afterShow': function () {

var rndColor = websocketclient.getRandomColor();

$("#colorChooser").minicolors('value', rndColor);

}

});

websocketclient.render.toggle('publish');

websocketclient.render.toggle('messages');

websocketclient.render.toggle('sub');

});

</script>

</body>

</html>

**Листинг style.css**

#header {

float: none;

margin: 0 auto;

padding: 0;

height: 80px;

background-color: #f2f2f2;

border-bottom: solid 2px #000000;

margin-bottom: 20px;

width: 100%;

background: #f2f2f2; /\* Old browsers \*/

background-image: -moz-linear-gradient(top, #f8f8f8 0%, #f2f2f2 100%); /\* FF3.6+ \*/

background-image: -webkit-gradient(linear, left top, left bottom, color-stop(0%, #f8f8f8), color-stop(100%, #f2f2f2)); /\* Chrome,Safari4+ \*/

background-image: -webkit-linear-gradient(top, #f8f8f8 0%, #f2f2f2 100%); /\* Chrome10+,Safari5.1+ \*/

background-image: -o-linear-gradient(top, #f8f8f8 0%, #f2f2f2 100%); /\* Opera 11.10+ \*/

background-image: -ms-linear-gradient(top, #f8f8f8 0%, #f2f2f2 100%); /\* IE10+ \*/

background-image: linear-gradient(to bottom, #f8f8f8 0%, #f2f2f2 100%); /\* W3C \*/

filter: progid:DXImageTransform.Microsoft.gradient(startColorstr='#f8f8f8', endColorstr='#f2f2f2', GradientType=0); /\* IE6-8 \*/

}

#header > div {

max-width: 62.5em;

margin: 0 auto;

}

#headertitle {

float: right;

margin-top: 30px;

margin-right: 20px;

}

#headerlogo {

margin-left: 20px;

margin-top: 5px;

}

.conniTop {

margin-left: 15px;

background-color: #f2f2f2;

width: 180px;

}

.publishTop {

background-color: #f2f2f2;

width: 160px;

}

.subTop {

background-color: #f2f2f2;

width: 215px;

}

.messagesTop {

background-color: #f2f2f2;

width: 180px;

}

div.conniArrow, div.publishArrow, div.subArrow, div.messagesArrow {

float: right;

text-align: right;

}

.checky {

margin-left: 22px;

}

.connection {

margin-bot: 20px;

}

.publish {

margin-top: 20px;

width: 60%;

}

.disc li {

list-style-type: none;

margin-bottom: 10px;

background-color: #ffffff;

}

#colorChooser {

width: 20px;

padding-left: 30px;

line-height: 24px;

}

/\*.subs0, .mess0{

border-left: solid 10px #ffff00;

background-color: #ffffff;

}

.subs1, .mess1{

border-left: solid 10px #ff00ff;

background-color: #ffffff;

}

.subs2, .mess2{

border-left: solid 10px #00ffff;

background-color: #ffffff;

}

.subs3, .mess3{

border-left: solid 10px #f0fff0;

background-color: #ffffff;

}

.subs4, .mess4{

border-left: solid 10px #ff0f0f;

background-color: #ffffff;

}\*/

.subText {

border-style: solid;

border-width: 1px;

border-color: #cccccc;

background-color: #ffffff;

text-overflow: ellipsis;

overflow: hidden;

white-space: nowrap;

}

.messageText {

border: solid 1px #cccccc;

}

.closer {

margin-bot: 20px;

padding: 0;

}

.left {

width: 10px;

background-color: #ffff00;

}

.right {

float: right;

color: #bababa;

}

.colorSub {

background-color: #ffff00;

}

.addSubButton {

width: 100%;

}

.empty {

clear: both;

height: 20px;

}

#connectButton, #disconnectButton {

margin-top: 17px;

}

#publishButton {

margin-top: 17px;

margin-left: -10px;

}

#filterButton {

margin-top: 17px;

}

#subscribeButton {

margin-top: 17px;

margin-left: -30px;

}

#lwQosInput {

margin-top: 7px;

margin-left: 20px;

width: 50px;

}

#publishQoSInput {

margin-top: 7px;

margin-left: -5px;

width: 50px;

}

#QoSInput {

margin-top: 7px;

margin-left: -5px;

width: 50px;

}

.qos, .date, .topicM, .retain {

color: #bababa;

font-size: 12px;

}

.columns h3 {

margin-bottom: 0;

margin-top: 0;

font-size: 23px;

}

form.custom .custom.dropdown.small {

width: 70px;

}

form.custom .custom.checkbox {

margin-left: 0.7em;

margin-top: 10px;

}

#connection, #publish-sub {

width: 100%;

}

.columns.subArrow, .columns.publishArrow, .columns.messagesArrow {

padding-right: 0;

}

#addSubButton {

margin-top: 15px;

}

#publishPanel {

padding-right: 0;

padding-left: 0;

}

div.messageText {

padding-left: 0;

padding-right: 0;

}

.row .messLine div.row, .row .subLine div.row {

margin-left: auto;

margin-right: auto;

}

#messagesMain {

margin-bottom: 20px;

}

div.icon-arrow-chevron {

background-image: url('../assets/arrow\_up.png');

min-height: 32px;

min-width: 32px;

background-repeat: no-repeat;

background-position: right;

}

.closed div.icon-arrow-chevron {

background-image: url('../assets/arrow\_down.png');

}

#connectionStatus {

margin-top: 7px;

width: 16px;

height: 16px;

border-radius: 8px;

}

body.notconnected #connectionStatus {

background-color: #bc0000;

}

body.connected #connectionStatus {

background-color: #009700;

}

body.notconnected #disconnectButton {

display: none;

}

body.connected #disconnectButton {

display: inline-block;

}

body.notconnected #connectButton {

display: inline-block;

}

body.connected #connectButton {

display: none;

}

body.connected #connectionStatus:after {

content: 'connected';

padding-left: 22px;

}

body.connectionbroke #connectionStatus:after {

content: 'disconnected';

padding-left: 22px;

}

.columns.subText {

padding-right: 2px;

}

.truncate {

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

}

.break-words {

word-wrap: break-word;

}

\*,

\*:before,

\*:after {

-moz-box-sizing: border-box;

-webkit-box-sizing: border-box;

box-sizing: border-box;

}

html,

body {

font-size: 100%;

}

body {

background: white;

color: #222222;

padding: 0;

margin: 0;

font-family: "Helvetica Neue", "Helvetica", Helvetica, Arial, sans-serif;

font-weight: normal;

font-style: normal;

line-height: 1;

position: relative;

cursor: default;

}

a:hover {

cursor: pointer;

}

a:focus {

outline: none;

}

img,

object,

embed {

max-width: 100%;

height: auto;

}

object,

embed {

height: 100%;

}

img {

-ms-interpolation-mode: bicubic;

}

#map\_canvas img,

#map\_canvas embed,

#map\_canvas object,

.map\_canvas img,

.map\_canvas embed,

.map\_canvas object {

max-width: none !important;

}

.left {

float: left !important;

}

.right {

float: right !important;

}

.text-left {

text-align: left !important;

}

.text-right {

text-align: right !important;

}

.text-center {

text-align: center !important;

}

.text-justify {

text-align: justify !important;

}

.hide {

display: none;

}

.antialiased {

-webkit-font-smoothing: antialiased;

}

img {

display: inline-block;

vertical-align: middle;

}

textarea {

height: auto;

min-height: 50px;

}

select {

width: 100%;

}

/\* Grid HTML Classes \*/

.row {

width: 100%;

margin-left: auto;

margin-right: auto;

margin-top: 0;

margin-bottom: 0;

max-width: 62.5em;

\*zoom: 1;

}

.row:before, .row:after {

content: " ";

display: table;

}

.row:after {

clear: both;

}

.row.collapse .column,

.row.collapse .columns {

position: relative;

padding-left: 0;

padding-right: 0;

float: left;

}

.row .row {

width: auto;

margin-left: -0.9375em;

margin-right: -0.9375em;

margin-top: 0;

margin-bottom: 0;

max-width: none;

\*zoom: 1;

}

.row .row:before, .row .row:after {

content: " ";

display: table;

}

.row .row:after {

clear: both;

}

.row .row.collapse {

width: auto;

margin: 0;

max-width: none;

\*zoom: 1;

}

.row .row.collapse:before, .row .row.collapse:after {

content: " ";

display: table;

}

.row .row.collapse:after {

clear: both;

}

.column,

.columns {

position: relative;

padding-left: 0.9375em;

padding-right: 0.9375em;

width: 100%;

float: left;

}

@media only screen {

.column,

.columns {

position: relative;

padding-left: 0.9375em;

padding-right: 0.9375em;

float: left;

}

.small-1 {

position: relative;

width: 8.33333%;

}

.small-2 {

position: relative;

width: 16.66667%;

}

.small-3 {

position: relative;

width: 25%;

}

.small-4 {

position: relative;

width: 33.33333%;

}

.small-5 {

position: relative;

width: 41.66667%;

}

.small-6 {

position: relative;

width: 50%;

}

.small-7 {

position: relative;

width: 58.33333%;

}

.small-8 {

position: relative;

width: 66.66667%;

}

.small-9 {

position: relative;

width: 75%;

}

.small-10 {

position: relative;

width: 83.33333%;

}

.small-11 {

position: relative;

width: 91.66667%;

}

.small-12 {

position: relative;

width: 100%;

}

.small-offset-0 {

position: relative;

margin-left: 0%;

}

.small-offset-1 {

position: relative;

margin-left: 8.33333%;

}

.small-offset-2 {

position: relative;

margin-left: 16.66667%;

}

.small-offset-3 {

position: relative;

margin-left: 25%;

}

.small-offset-4 {

position: relative;

margin-left: 33.33333%;

}

.small-offset-5 {

position: relative;

margin-left: 41.66667%;

}

.small-offset-6 {

position: relative;

margin-left: 50%;

}

.small-offset-7 {

position: relative;

margin-left: 58.33333%;

}

.small-offset-8 {

position: relative;

margin-left: 66.66667%;

}

.small-offset-9 {

position: relative;

margin-left: 75%;

}

.small-offset-10 {

position: relative;

margin-left: 83.33333%;

}

[class\*="column"] + [class\*="column"]:last-child {

float: right;

}

[class\*="column"] + [class\*="column"].end {

float: left;

}

.column.small-centered,

.columns.small-centered {

position: relative;

margin-left: auto;

margin-right: auto;

float: none !important;

}

}

/\* Styles for screens that are atleast 768px; \*/

@media only screen and (min-width: 768px) {

.large-1 {

position: relative;

width: 8.33333%;

}

.large-2 {

position: relative;

width: 16.66667%;

}

.large-3 {

position: relative;

width: 25%;

}

.large-4 {

position: relative;

width: 33.33333%;

}

.large-5 {

position: relative;

width: 41.66667%;

}

.large-6 {

position: relative;

width: 50%;

}

.large-7 {

position: relative;

width: 58.33333%;

}

.large-8 {

position: relative;

width: 66.66667%;

}

.large-9 {

position: relative;

width: 75%;

}

.large-10 {

position: relative;

width: 83.33333%;

}

.large-11 {

position: relative;

width: 91.66667%;

}

.large-12 {

position: relative;

width: 100%;

}

.row .large-offset-0 {

position: relative;

margin-left: 0%;

}

.row .large-offset-1 {

position: relative;

margin-left: 8.33333%;

}

.row .large-offset-2 {

position: relative;

margin-left: 16.66667%;

}

.row .large-offset-3 {

position: relative;

margin-left: 25%;

}

.row .large-offset-4 {

position: relative;

margin-left: 33.33333%;

}

.row .large-offset-5 {

position: relative;

margin-left: 41.66667%;

}

.row .large-offset-6 {

position: relative;

margin-left: 50%;

}

.row .large-offset-7 {

position: relative;

margin-left: 58.33333%;

}

.row .large-offset-8 {

position: relative;

margin-left: 66.66667%;

}

.row .large-offset-9 {

position: relative;

margin-left: 75%;

}

.row .large-offset-10 {

position: relative;

margin-left: 83.33333%;

}

.row .large-offset-11 {

position: relative;

margin-left: 91.66667%;

}

.push-1 {

position: relative;

left: 8.33333%;

right: auto;

}

.pull-1 {

position: relative;

right: 8.33333%;

left: auto;

}

.push-2 {

position: relative;

left: 16.66667%;

right: auto;

}

.pull-2 {

position: relative;

right: 16.66667%;

left: auto;

}

.push-3 {

position: relative;

left: 25%;

right: auto;

}

.pull-3 {

position: relative;

right: 25%;

left: auto;

}

.push-4 {

position: relative;

left: 33.33333%;

right: auto;

}

.pull-4 {

position: relative;

right: 33.33333%;

left: auto;

}

.push-5 {

position: relative;

left: 41.66667%;

right: auto;

}

.pull-5 {

position: relative;

right: 41.66667%;

left: auto;

}

.push-6 {

position: relative;

left: 50%;

right: auto;

}

.pull-6 {

position: relative;

right: 50%;

left: auto;

}

.push-7 {

position: relative;

left: 58.33333%;

right: auto;

}

.pull-7 {

position: relative;

right: 58.33333%;

left: auto;

}

.push-8 {

position: relative;

left: 66.66667%;

right: auto;

}

.pull-8 {

position: relative;

right: 66.66667%;

left: auto;

}

.push-9 {

position: relative;

left: 75%;

right: auto;

}

.pull-9 {

position: relative;

right: 75%;

left: auto;

}

.push-10 {

position: relative;

left: 83.33333%;

right: auto;

}

.pull-10 {

position: relative;

right: 83.33333%;

left: auto;

}

.push-11 {

position: relative;

left: 91.66667%;

right: auto;

}

.pull-11 {

position: relative;

right: 91.66667%;

left: auto;

}

.column.large-centered,

.columns.large-centered {

position: relative;

margin-left: auto;

margin-right: auto;

float: none !important;

}

.column.large-uncentered,

.columns.large-uncentered {

margin-left: 0;

margin-right: 0;

float: left !important;

}

.column.large-uncentered.opposite,

.columns.large-uncentered.opposite {

float: right !important;

}

}

/\* Foundation Visibility HTML Classes \*/

.show-for-small,

.show-for-medium-down,

.show-for-large-down {

display: inherit !important;

}

.show-for-medium,

.show-for-medium-up,

.show-for-large,

.show-for-large-up,

.show-for-xlarge {

display: none !important;

}

.hide-for-medium,

.hide-for-medium-up,

.hide-for-large,

.hide-for-large-up,

.hide-for-xlarge {

display: inherit !important;

}

.hide-for-small,

.hide-for-medium-down,

.hide-for-large-down {

display: none !important;

}

/\* Specific visilbity for tables \*/

table.show-for-small, table.show-for-medium-down, table.show-for-large-down, table.hide-for-medium, table.hide-for-medium-up, table.hide-for-large, table.hide-for-large-up, table.hide-for-xlarge {

display: table;

}

thead.show-for-small, thead.show-for-medium-down, thead.show-for-large-down, thead.hide-for-medium, thead.hide-for-medium-up, thead.hide-for-large, thead.hide-for-large-up, thead.hide-for-xlarge {

display: table-header-group !important;

}

tbody.show-for-small, tbody.show-for-medium-down, tbody.show-for-large-down, tbody.hide-for-medium, tbody.hide-for-medium-up, tbody.hide-for-large, tbody.hide-for-large-up, tbody.hide-for-xlarge {

display: table-row-group !important;

}

tr.show-for-small, tr.show-for-medium-down, tr.show-for-large-down, tr.hide-for-medium, tr.hide-for-medium-up, tr.hide-for-large, tr.hide-for-large-up, tr.hide-for-xlarge {

display: table-row !important;

}

td.show-for-small, td.show-for-medium-down, td.show-for-large-down, td.hide-for-medium, td.hide-for-medium-up, td.hide-for-large, td.hide-for-large-up, td.hide-for-xlarge,

th.show-for-small,

th.show-for-medium-down,

th.show-for-large-down,

th.hide-for-medium,

th.hide-for-medium-up,

th.hide-for-large,

th.hide-for-large-up,

th.hide-for-xlarge {

display: table-cell !important;

}

/\* Medium Displays: 768px - 1279px \*/

@media only screen and (min-width: 768px) {

.show-for-medium,

.show-for-medium-up {

display: inherit !important;

}

.show-for-small {

display: none !important;

}

.hide-for-small {

display: inherit !important;

}

.hide-for-medium,

.hide-for-medium-up {

display: none !important;

}

/\* Specific visilbity for tables \*/

table.show-for-medium, table.show-for-medium-up, table.hide-for-small {

display: table;

}

thead.show-for-medium, thead.show-for-medium-up, thead.hide-for-small {

display: table-header-group !important;

}

tbody.show-for-medium, tbody.show-for-medium-up, tbody.hide-for-small {

display: table-row-group !important;

}

tr.show-for-medium, tr.show-for-medium-up, tr.hide-for-small {

display: table-row !important;

}

td.show-for-medium, td.show-for-medium-up, td.hide-for-small,

th.show-for-medium,

th.show-for-medium-up,

th.hide-for-small {

display: table-cell !important;

}

}

/\* Large Displays: 1280px - 1440px \*/

@media only screen and (min-width: 1280px) {

.show-for-large,

.show-for-large-up {

display: inherit !important;

}

.show-for-medium,

.show-for-medium-down {

display: none !important;

}

.hide-for-medium,

.hide-for-medium-down {

display: inherit !important;

}

.hide-for-large,

.hide-for-large-up {

display: none !important;

}

/\* Specific visilbity for tables \*/

table.show-for-large, table.show-for-large-up, table.hide-for-medium, table.hide-for-medium-down {

display: table;

}

thead.show-for-large, thead.show-for-large-up, thead.hide-for-medium, thead.hide-for-medium-down {

display: table-header-group !important;

}

tbody.show-for-large, tbody.show-for-large-up, tbody.hide-for-medium, tbody.hide-for-medium-down {

display: table-row-group !important;

}

tr.show-for-large, tr.show-for-large-up, tr.hide-for-medium, tr.hide-for-medium-down {

display: table-row !important;

}

td.show-for-large, td.show-for-large-up, td.hide-for-medium, td.hide-for-medium-down,

th.show-for-large,

th.show-for-large-up,

th.hide-for-medium,

th.hide-for-medium-down {

display: table-cell !important;

}

}

/\* X-Large Displays: 1400px and up \*/

@media only screen and (min-width: 1440px) {

.show-for-xlarge {

display: inherit !important;

}

.show-for-large,

.show-for-large-down {

display: none !important;

}

.hide-for-large,

.hide-for-large-down {

display: inherit !important;

}

.hide-for-xlarge {

display: none !important;

}

/\* Specific visilbity for tables \*/

table.show-for-xlarge, table.hide-for-large, table.hide-for-large-down {

display: table;

}

thead.show-for-xlarge, thead.hide-for-large, thead.hide-for-large-down {

display: table-header-group !important;

}

tbody.show-for-xlarge, tbody.hide-for-large, tbody.hide-for-large-down {

display: table-row-group !important;

}

tr.show-for-xlarge, tr.hide-for-large, tr.hide-for-large-down {

display: table-row !important;

}

td.show-for-xlarge, td.hide-for-large, td.hide-for-large-down,

th.show-for-xlarge,

th.hide-for-large,

th.hide-for-large-down {

display: table-cell !important;

}

}

/\* Orientation targeting \*/

.show-for-landscape,

.hide-for-portrait {

display: inherit !important;

}

.hide-for-landscape,

.show-for-portrait {

display: none !important;

}

/\* Specific visilbity for tables \*/

table.hide-for-landscape, table.show-for-portrait {

display: table;

}

thead.hide-for-landscape, thead.show-for-portrait {

display: table-header-group !important;

}

tbody.hide-for-landscape, tbody.show-for-portrait {

display: table-row-group !important;

}

tr.hide-for-landscape, tr.show-for-portrait {

display: table-row !important;

}

td.hide-for-landscape, td.show-for-portrait,

th.hide-for-landscape,

th.show-for-portrait {

display: table-cell !important;

}

@media only screen and (orientation: landscape) {

.show-for-landscape,

.hide-for-portrait {

display: inherit !important;

}

.hide-for-landscape,

.show-for-portrait {

display: none !important;

}

/\* Specific visilbity for tables \*/

table.show-for-landscape, table.hide-for-portrait {

display: table;

}

thead.show-for-landscape, thead.hide-for-portrait {

display: table-header-group !important;

}

tbody.show-for-landscape, tbody.hide-for-portrait {

display: table-row-group !important;

}

tr.show-for-landscape, tr.hide-for-portrait {

display: table-row !important;

}

td.show-for-landscape, td.hide-for-portrait,

th.show-for-landscape,

th.hide-for-portrait {

display: table-cell !important;

}

}

@media only screen and (orientation: portrait) {

.show-for-portrait,

.hide-for-landscape {

display: inherit !important;

}

.hide-for-portrait,

.show-for-landscape {

display: none !important;

}

/\* Specific visilbity for tables \*/

table.show-for-portrait, table.hide-for-landscape {

display: table;

}

thead.show-for-portrait, thead.hide-for-landscape {

display: table-header-group !important;

}

tbody.show-for-portrait, tbody.hide-for-landscape {

display: table-row-group !important;

}

tr.show-for-portrait, tr.hide-for-landscape {

display: table-row !important;

}

td.show-for-portrait, td.hide-for-landscape,

th.show-for-portrait,

th.hide-for-landscape {

display: table-cell !important;

}

}

/\* Touch-enabled device targeting \*/

.show-for-touch {

display: none !important;

}

.hide-for-touch {

display: inherit !important;

}

.touch .show-for-touch {

display: inherit !important;

}

.touch .hide-for-touch {

display: none !important;

}

/\* Specific visilbity for tables \*/

table.hide-for-touch {

display: table;

}

.touch table.show-for-touch {

display: table;

}

thead.hide-for-touch {

display: table-header-group !important;

}

.touch thead.show-for-touch {

display: table-header-group !important;

}

tbody.hide-for-touch {

display: table-row-group !important;

}

.touch tbody.show-for-touch {

display: table-row-group !important;

}

tr.hide-for-touch {

display: table-row !important;

}

.touch tr.show-for-touch {

display: table-row !important;

}

td.hide-for-touch {

display: table-cell !important;

}

.touch td.show-for-touch {

display: table-cell !important;

}

th.hide-for-touch {

display: table-cell !important;

}

.touch th.show-for-touch {

display: table-cell !important;

}

/\* Foundation Block Grids for below small breakpoint \*/

@media only screen {

[class\*="block-grid-"] {

display: block;

padding: 0;

margin: 0 -0.625em;

\*zoom: 1;

}

[class\*="block-grid-"]:before, [class\*="block-grid-"]:after {

content: " ";

display: table;

}

[class\*="block-grid-"]:after {

clear: both;

}

[class\*="block-grid-"] > li {

display: inline;

height: auto;

float: left;

padding: 0 0.625em 1.25em;

}

.small-block-grid-1 > li {

width: 100%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-1 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-1 > li:nth-of-type(1n+1) {

clear: both;

}

.small-block-grid-2 > li {

width: 50%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-2 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-2 > li:nth-of-type(2n+1) {

clear: both;

}

.small-block-grid-3 > li {

width: 33.33333%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-3 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-3 > li:nth-of-type(3n+1) {

clear: both;

}

.small-block-grid-4 > li {

width: 25%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-4 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-4 > li:nth-of-type(4n+1) {

clear: both;

}

.small-block-grid-5 > li {

width: 20%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-5 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-5 > li:nth-of-type(5n+1) {

clear: both;

}

.small-block-grid-6 > li {

width: 16.66667%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-6 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-6 > li:nth-of-type(6n+1) {

clear: both;

}

.small-block-grid-7 > li {

width: 14.28571%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-7 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-7 > li:nth-of-type(7n+1) {

clear: both;

}

.small-block-grid-8 > li {

width: 12.5%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-8 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-8 > li:nth-of-type(8n+1) {

clear: both;

}

.small-block-grid-9 > li {

width: 11.11111%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-9 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-9 > li:nth-of-type(9n+1) {

clear: both;

}

.small-block-grid-10 > li {

width: 10%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-10 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-10 > li:nth-of-type(10n+1) {

clear: both;

}

.small-block-grid-11 > li {

width: 9.09091%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-11 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-11 > li:nth-of-type(11n+1) {

clear: both;

}

.small-block-grid-12 > li {

width: 8.33333%;

padding: 0 0.625em 1.25em;

}

.small-block-grid-12 > li:nth-of-type(n) {

clear: none;

}

.small-block-grid-12 > li:nth-of-type(12n+1) {

clear: both;

}

}

/\* Foundation Block Grids for above small breakpoint \*/

@media only screen and (min-width: 768px) {

/\* Remove small grid clearing \*/

.small-block-grid-1 > li:nth-of-type(1n+1) {

clear: none;

}

.small-block-grid-2 > li:nth-of-type(2n+1) {

clear: none;

}

.small-block-grid-3 > li:nth-of-type(3n+1) {

clear: none;

}

.small-block-grid-4 > li:nth-of-type(4n+1) {

clear: none;

}

.small-block-grid-5 > li:nth-of-type(5n+1) {

clear: none;

}

.small-block-grid-6 > li:nth-of-type(6n+1) {

clear: none;

}

.small-block-grid-7 > li:nth-of-type(7n+1) {

clear: none;

}

.small-block-grid-8 > li:nth-of-type(8n+1) {

clear: none;

}

.small-block-grid-9 > li:nth-of-type(9n+1) {

clear: none;

}

.small-block-grid-10 > li:nth-of-type(10n+1) {

clear: none;

}

.small-block-grid-11 > li:nth-of-type(11n+1) {

clear: none;

}

.small-block-grid-12 > li:nth-of-type(12n+1) {

clear: none;

}

.large-block-grid-1 > li {

width: 100%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-1 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-1 > li:nth-of-type(1n+1) {

clear: both;

}

.large-block-grid-2 > li {

width: 50%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-2 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-2 > li:nth-of-type(2n+1) {

clear: both;

}

.large-block-grid-3 > li {

width: 33.33333%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-3 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-3 > li:nth-of-type(3n+1) {

clear: both;

}

.large-block-grid-4 > li {

width: 25%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-4 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-4 > li:nth-of-type(4n+1) {

clear: both;

}

.large-block-grid-5 > li {

width: 20%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-5 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-5 > li:nth-of-type(5n+1) {

clear: both;

}

.large-block-grid-6 > li {

width: 16.66667%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-6 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-6 > li:nth-of-type(6n+1) {

clear: both;

}

.large-block-grid-7 > li {

width: 14.28571%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-7 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-7 > li:nth-of-type(7n+1) {

clear: both;

}

.large-block-grid-8 > li {

width: 12.5%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-8 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-8 > li:nth-of-type(8n+1) {

clear: both;

}

.large-block-grid-9 > li {

width: 11.11111%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-9 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-9 > li:nth-of-type(9n+1) {

clear: both;

}

.large-block-grid-10 > li {

width: 10%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-10 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-10 > li:nth-of-type(10n+1) {

clear: both;

}

.large-block-grid-11 > li {

width: 9.09091%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-11 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-11 > li:nth-of-type(11n+1) {

clear: both;

}

.large-block-grid-12 > li {

width: 8.33333%;

padding: 0 0.625em 1.25em;

}

.large-block-grid-12 > li:nth-of-type(n) {

clear: none;

}

.large-block-grid-12 > li:nth-of-type(12n+1) {

clear: both;

}

}

p.lead {

font-size: 1.21875em;

line-height: 1.6;

}

.subheader {

line-height: 1.4;

color: #6f6f6f;

font-weight: 300;

margin-top: 0.2em;

margin-bottom: 0.5em;

}

/\* Typography resets \*/

div,

dl,

dt,

dd,

ul,

ol,

li,

h1,

h2,

h3,

h4,

h5,

h6,

pre,

form,

p,

blockquote,

th,

td {

margin: 0;

padding: 0;

direction: ltr;

}

/\* Default Link Styles \*/

a {

color: #2ba6cb;

text-decoration: none;

line-height: inherit;

}

a:hover, a:focus {

color: #2795b6;

}

a img {

border: none;

}

/\* Default paragraph styles \*/

p {

font-family: inherit;

font-weight: normal;

font-size: 1em;

line-height: 1.6;

margin-bottom: 1.25em;

text-rendering: optimizeLegibility;

}

p aside {

font-size: 0.875em;

line-height: 1.35;

font-style: italic;

}

/\* Default header styles \*/

h1, h2, h3, h4, h5, h6 {

font-family: "Helvetica Neue", "Helvetica", Helvetica, Arial, sans-serif;

font-weight: bold;

font-style: normal;

color: #222222;

text-rendering: optimizeLegibility;

margin-top: 0.2em;

margin-bottom: 0.5em;

line-height: 1.2125em;

}

h1 small, h2 small, h3 small, h4 small, h5 small, h6 small {

font-size: 60%;

color: #6f6f6f;

line-height: 0;

}

h1 {

font-size: 2.125em;

}

h2 {

font-size: 1.6875em;

}

h3 {

font-size: 1.375em;

}

h4 {

font-size: 1.125em;

}

h5 {

font-size: 1.125em;

}

h6 {

font-size: 1em;

}

hr {

border: solid #dddddd;

border-width: 1px 0 0;

clear: both;

margin: 1.25em 0 1.1875em;

height: 0;

}

/\* Helpful Typography Defaults \*/

em,

i {

font-style: italic;

line-height: inherit;

}

strong,

b {

font-weight: bold;

line-height: inherit;

}

small {

font-size: 60%;

line-height: inherit;

}

code {

font-family: Consolas, "Liberation Mono", Courier, monospace;

font-weight: bold;

color: #7f0a0c;

}

/\* Lists \*/

ul,

ol,

dl {

font-size: 1em;

line-height: 1.6;

margin-bottom: 1.25em;

list-style-position: outside;

font-family: inherit;

}

ul, ol {

margin-left: 0;

}

/\* Unordered Lists \*/

ul li ul,

ul li ol {

margin-left: 1.25em;

margin-bottom: 0;

font-size: 1em;

/\* Override nested font-size change \*/

}

ul.square li ul, ul.circle li ul, ul.disc li ul {

list-style: inherit;

}

ul.square {

list-style-type: square;

}

ul.circle {

list-style-type: circle;

}

ul.disc {

list-style-type: disc;

}

ul.no-bullet {

list-style: none;

}

/\* Ordered Lists \*/

ol li ul,

ol li ol {

margin-left: 1.25em;

margin-bottom: 0;

}

/\* Definition Lists \*/

dl dt {

margin-bottom: 0.3em;

font-weight: bold;

}

dl dd {

margin-bottom: 0.75em;

}

/\* Abbreviations \*/

abbr,

acronym {

text-transform: uppercase;

font-size: 90%;

color: #222222;

border-bottom: 1px dotted #dddddd;

cursor: help;

}

abbr {

text-transform: none;

}

/\* Blockquotes \*/

blockquote {

margin: 0 0 1.25em;

padding: 0.5625em 1.25em 0 1.1875em;

border-left: 1px solid #dddddd;

}

blockquote cite {

display: block;

font-size: 0.8125em;

color: #555555;

}

blockquote cite:before {

content: "\2014 \0020";

}

blockquote cite a,

blockquote cite a:visited {

color: #555555;

}

blockquote,

blockquote p {

line-height: 1.6;

color: #6f6f6f;

}

/\* Microformats \*/

.vcard {

display: inline-block;

margin: 0 0 1.25em 0;

border: 1px solid #dddddd;

padding: 0.625em 0.75em;

}

.vcard li {

margin: 0;

display: block;

}

.vcard .fn {

font-weight: bold;

font-size: 0.9375em;

}

.vevent .summary {

font-weight: bold;

}

.vevent abbr {

cursor: default;

text-decoration: none;

font-weight: bold;

border: none;

padding: 0 0.0625em;

}

@media only screen and (min-width: 768px) {

h1, h2, h3, h4, h5, h6 {

line-height: 1.4;

}

h1 {

font-size: 2.75em;

}

h2 {

font-size: 2.3125em;

}

h3 {

font-size: 1.6875em;

}

h4 {

font-size: 1.4375em;

}

}

/\*

\* Print styles.

\*

\* Inlined to avoid required HTTP connection: www.phpied.com/delay-loading-your-print-css/

\* Credit to Paul Irish and HTML5 Boilerplate (html5boilerplate.com)

\*/

.print-only {

display: none !important;

}

@media print {

\* {

background: transparent !important;

color: black !important;

/\* Black prints faster: h5bp.com/s \*/

box-shadow: none !important;

text-shadow: none !important;

}

a,

a:visited {

text-decoration: underline;

}

a[href]:after {

content: " (" attr(href) ")";

}

abbr[title]:after {

content: " (" attr(title) ")";

}

.ir a:after,

a[href^="javascript:"]:after,

a[href^="#"]:after {

content: "";

}

pre,

blockquote {

border: 1px solid #999999;

page-break-inside: avoid;

}

thead {

display: table-header-group;

/\* h5bp.com/t \*/

}

tr,

img {

page-break-inside: avoid;

}

img {

max-width: 100% !important;

}

@page {

margin: 0.5cm;

}

p,

h2,

h3 {

orphans: 3;

widows: 3;

}

h2,

h3 {

page-break-after: avoid;

}

.hide-on-print {

display: none !important;

}

.print-only {

display: block !important;

}

.hide-for-print {

display: none !important;

}

.show-for-print {

display: inherit !important;

}

}

button, .button {

border-style: solid;

border-width: 1px;

cursor: pointer;

font-family: inherit;

font-weight: bold;

line-height: 1;

margin: 0 0 1.25em;

position: relative;

text-decoration: none;

text-align: center;

display: inline-block;

padding-top: 0.75em;

padding-right: 1.5em;

padding-bottom: 0.8125em;

padding-left: 1.5em;

font-size: 1em;

background-color: #2ba6cb;

border-color: #2284a1;

color: white;

}

button:hover, button:focus, .button:hover, .button:focus {

background-color: #2284a1;

}

button:hover, button:focus, .button:hover, .button:focus {

color: white;

}

button.secondary, .button.secondary {

background-color: #e9e9e9;

border-color: #d0d0d0;

color: #333333;

}

button.secondary:hover, button.secondary:focus, .button.secondary:hover, .button.secondary:focus {

background-color: #d0d0d0;

}

button.secondary:hover, button.secondary:focus, .button.secondary:hover, .button.secondary:focus {

color: #333333;

}

button.success, .button.success {

background-color: #5da423;

border-color: #457a1a;

color: white;

}

button.success:hover, button.success:focus, .button.success:hover, .button.success:focus {

background-color: #457a1a;

}

button.success:hover, button.success:focus, .button.success:hover, .button.success:focus {

color: white;

}

button.alert, .button.alert {

background-color: #c60f13;

border-color: #970b0e;

color: white;

}

button.alert:hover, button.alert:focus, .button.alert:hover, .button.alert:focus {

background-color: #970b0e;

}

button.alert:hover, button.alert:focus, .button.alert:hover, .button.alert:focus {

color: white;

}

button.large, .button.large {

padding-top: 1em;

padding-right: 2em;

padding-bottom: 1.0625em;

padding-left: 2em;

font-size: 1.25em;

}

button.small, .button.small {

padding-top: 0.5625em;

padding-right: 1.125em;

padding-bottom: 0.625em;

padding-left: 1.125em;

font-size: 0.8125em;

}

button.tiny, .button.tiny {

padding-top: 0.4375em;

padding-right: 0.875em;

padding-bottom: 0.5em;

padding-left: 0.875em;

font-size: 0.6875em;

}

button.expand, .button.expand {

padding-right: 0px;

padding-left: 0px;

width: 100%;

}

button.left-align, .button.left-align {

text-align: left;

text-indent: 0.75em;

}

button.right-align, .button.right-align {

text-align: right;

padding-right: 0.75em;

}

button.disabled, button[disabled], .button.disabled, .button[disabled] {

background-color: #2ba6cb;

border-color: #2284a1;

color: white;

cursor: default;

opacity: 0.6;

-webkit-box-shadow: none;

box-shadow: none;

}

button.disabled:hover, button.disabled:focus, button[disabled]:hover, button[disabled]:focus, .button.disabled:hover, .button.disabled:focus, .button[disabled]:hover, .button[disabled]:focus {

background-color: #2284a1;

}

button.disabled:hover, button.disabled:focus, button[disabled]:hover, button[disabled]:focus, .button.disabled:hover, .button.disabled:focus, .button[disabled]:hover, .button[disabled]:focus {

color: white;

}

button.disabled:hover, button.disabled:focus, button[disabled]:hover, button[disabled]:focus, .button.disabled:hover, .button.disabled:focus, .button[disabled]:hover, .button[disabled]:focus {

background-color: #2ba6cb;

}

button.disabled.secondary, button[disabled].secondary, .button.disabled.secondary, .button[disabled].secondary {

background-color: #e9e9e9;

border-color: #d0d0d0;

color: #333333;

cursor: default;

opacity: 0.6;

-webkit-box-shadow: none;

box-shadow: none;

}

button.disabled.secondary:hover, button.disabled.secondary:focus, button[disabled].secondary:hover, button[disabled].secondary:focus, .button.disabled.secondary:hover, .button.disabled.secondary:focus, .button[disabled].secondary:hover, .button[disabled].secondary:focus {

background-color: #d0d0d0;

}

button.disabled.secondary:hover, button.disabled.secondary:focus, button[disabled].secondary:hover, button[disabled].secondary:focus, .button.disabled.secondary:hover, .button.disabled.secondary:focus, .button[disabled].secondary:hover, .button[disabled].secondary:focus {

color: #333333;

}

button.disabled.secondary:hover, button.disabled.secondary:focus, button[disabled].secondary:hover, button[disabled].secondary:focus, .button.disabled.secondary:hover, .button.disabled.secondary:focus, .button[disabled].secondary:hover, .button[disabled].secondary:focus {

background-color: #e9e9e9;

}

button.disabled.success, button[disabled].success, .button.disabled.success, .button[disabled].success {

background-color: #5da423;

border-color: #457a1a;

color: white;

cursor: default;

opacity: 0.6;

-webkit-box-shadow: none;

box-shadow: none;

}

button.disabled.success:hover, button.disabled.success:focus, button[disabled].success:hover, button[disabled].success:focus, .button.disabled.success:hover, .button.disabled.success:focus, .button[disabled].success:hover, .button[disabled].success:focus {

background-color: #457a1a;

}

button.disabled.success:hover, button.disabled.success:focus, button[disabled].success:hover, button[disabled].success:focus, .button.disabled.success:hover, .button.disabled.success:focus, .button[disabled].success:hover, .button[disabled].success:focus {

color: white;

}

button.disabled.success:hover, button.disabled.success:focus, button[disabled].success:hover, button[disabled].success:focus, .button.disabled.success:hover, .button.disabled.success:focus, .button[disabled].success:hover, .button[disabled].success:focus {

background-color: #5da423;

}

button.disabled.alert, button[disabled].alert, .button.disabled.alert, .button[disabled].alert {

background-color: #c60f13;

border-color: #970b0e;

color: white;

cursor: default;

opacity: 0.6;

-webkit-box-shadow: none;

box-shadow: none;

}

button.disabled.alert:hover, button.disabled.alert:focus, button[disabled].alert:hover, button[disabled].alert:focus, .button.disabled.alert:hover, .button.disabled.alert:focus, .button[disabled].alert:hover, .button[disabled].alert:focus {

background-color: #970b0e;

}

button.disabled.alert:hover, button.disabled.alert:focus, button[disabled].alert:hover, button[disabled].alert:focus, .button.disabled.alert:hover, .button.disabled.alert:focus, .button[disabled].alert:hover, .button[disabled].alert:focus {

color: white;

}

button.disabled.alert:hover, button.disabled.alert:focus, button[disabled].alert:hover, button[disabled].alert:focus, .button.disabled.alert:hover, .button.disabled.alert:focus, .button[disabled].alert:hover, .button[disabled].alert:focus {

background-color: #c60f13;

}

button, .button {

padding-top: 0.8125em;

padding-bottom: 0.75em;

-webkit-appearance: none;

}

button.tiny, .button.tiny {

padding-top: 0.5em;

padding-bottom: 0.4375em;

-webkit-appearance: none;

}

button.small, .button.small {

padding-top: 0.625em;

padding-bottom: 0.5625em;

-webkit-appearance: none;

}

button.large, .button.large {

padding-top: 1.03125em;

padding-bottom: 1.03125em;

-webkit-appearance: none;

}

@media only screen {

button, .button {

-webkit-box-shadow: 0 1px 0 rgba(255, 255, 255, 0.5) inset;

box-shadow: 0 1px 0 rgba(255, 255, 255, 0.5) inset;

-webkit-transition: background-color 300ms ease-out;

-moz-transition: background-color 300ms ease-out;

transition: background-color 300ms ease-out;

}

button:active, .button:active {

-webkit-box-shadow: 0 1px 0 rgba(0, 0, 0, 0.2) inset;

box-shadow: 0 1px 0 rgba(0, 0, 0, 0.2) inset;

}

button.radius, .button.radius {

-webkit-border-radius: 3px;

border-radius: 3px;

}

button.round, .button.round {

-webkit-border-radius: 1000px;

border-radius: 1000px;

}

}

@media only screen and (min-width: 768px) {

button, .button {

display: inline-block;

}

}

/\* Standard Forms \*/

form {

margin: 0 0 1em;

}

/\* Using forms within rows, we need to set some defaults \*/

form .row .row {

margin: 0 -0.5em;

}

form .row .row .column,

form .row .row .columns {

padding: 0 0.5em;

}

form .row .row.collapse {

margin: 0;

}

form .row .row.collapse .column,

form .row .row.collapse .columns {

padding: 0;

}

form .row input.column,

form .row input.columns,

form .row textarea.column,

form .row textarea.columns {

padding-left: 0.5em;

}

/\* Label Styles \*/

label {

font-size: 0.875em;

color: #4d4d4d;

cursor: pointer;

display: block;

font-weight: 500;

margin-bottom: 0.1875em;

/\* Styles for required inputs \*/

}

label.right {

float: none;

text-align: right;

}

label.inline {

margin: 0 0 1em 0;

padding: 0.625em 0;

}

label small {

text-transform: capitalize;

color: #666666;

}

/\* Attach elements to the beginning or end of an input \*/

.prefix,

.postfix {

display: block;

position: relative;

z-index: 2;

text-align: center;

width: 100%;

padding-top: 0;

padding-bottom: 0;

border-style: solid;

border-width: 1px;

overflow: hidden;

font-size: 0.875em;

height: 2.3125em;

line-height: 2.3125em;

}

/\* Adjust padding, alignment and radius if pre/post element is a button \*/

.postfix.button {

padding-left: 0;

padding-right: 0;

padding-top: 0;

padding-bottom: 0;

text-align: center;

line-height: 2.125em;

}

.prefix.button {

padding-left: 0;

padding-right: 0;

padding-top: 0;

padding-bottom: 0;

text-align: center;

line-height: 2.125em;

}

.prefix.button.radius {

-webkit-border-radius: 0;

border-radius: 0;

-moz-border-radius-bottomleft: 3px;

-moz-border-radius-topleft: 3px;

-webkit-border-bottom-left-radius: 3px;

-webkit-border-top-left-radius: 3px;

border-bottom-left-radius: 3px;

border-top-left-radius: 3px;

}

.postfix.button.radius {

-webkit-border-radius: 0;

border-radius: 0;

-moz-border-radius-topright: 3px;

-moz-border-radius-bottomright: 3px;

-webkit-border-top-right-radius: 3px;

-webkit-border-bottom-right-radius: 3px;

border-top-right-radius: 3px;

border-bottom-right-radius: 3px;

}

.prefix.button.round {

-webkit-border-radius: 0;

border-radius: 0;

-moz-border-radius-bottomleft: 1000px;

-moz-border-radius-topleft: 1000px;

-webkit-border-bottom-left-radius: 1000px;

-webkit-border-top-left-radius: 1000px;

border-bottom-left-radius: 1000px;

border-top-left-radius: 1000px;

}

.postfix.button.round {

-webkit-border-radius: 0;

border-radius: 0;

-moz-border-radius-topright: 1000px;

-moz-border-radius-bottomright: 1000px;

-webkit-border-top-right-radius: 1000px;

-webkit-border-bottom-right-radius: 1000px;

border-top-right-radius: 1000px;

border-bottom-right-radius: 1000px;

}

/\* Separate prefix and postfix styles when on span or label so buttons keep their own \*/

span.prefix, label.prefix {

background: #f2f2f2;

border-color: #d9d9d9;

border-right: none;

color: #333333;

}

span.prefix.radius, label.prefix.radius {

-webkit-border-radius: 0;

border-radius: 0;

-moz-border-radius-bottomleft: 3px;

-moz-border-radius-topleft: 3px;

-webkit-border-bottom-left-radius: 3px;

-webkit-border-top-left-radius: 3px;

border-bottom-left-radius: 3px;

border-top-left-radius: 3px;

}

span.postfix, label.postfix {

background: #f2f2f2;

border-color: #cccccc;

border-left: none;

color: #333333;

}

span.postfix.radius, label.postfix.radius {

-webkit-border-radius: 0;

border-radius: 0;

-moz-border-radius-topright: 3px;

-moz-border-radius-bottomright: 3px;

-webkit-border-top-right-radius: 3px;

-webkit-border-bottom-right-radius: 3px;

border-top-right-radius: 3px;

border-bottom-right-radius: 3px;

}

/\* Input groups will automatically style first and last elements of the group \*/

.input-group.radius > \*:first-child, .input-group.radius > \*:first-child \* {

-moz-border-radius-bottomleft: 3px;

-moz-border-radius-topleft: 3px;

-webkit-border-bottom-left-radius: 3px;

-webkit-border-top-left-radius: 3px;

border-bottom-left-radius: 3px;

border-top-left-radius: 3px;

}

.input-group.radius > \*:last-child, .input-group.radius > \*:last-child \* {

-moz-border-radius-topright: 3px;

-moz-border-radius-bottomright: 3px;

-webkit-border-top-right-radius: 3px;

-webkit-border-bottom-right-radius: 3px;

border-top-right-radius: 3px;

border-bottom-right-radius: 3px;

}

.input-group.round > \*:first-child, .input-group.round > \*:first-child \* {

-moz-border-radius-bottomleft: 1000px;

-moz-border-radius-topleft: 1000px;

-webkit-border-bottom-left-radius: 1000px;

-webkit-border-top-left-radius: 1000px;

border-bottom-left-radius: 1000px;

border-top-left-radius: 1000px;

}

.input-group.round > \*:last-child, .input-group.round > \*:last-child \* {

-moz-border-radius-topright: 1000px;

-moz-border-radius-bottomright: 1000px;

-webkit-border-top-right-radius: 1000px;

-webkit-border-bottom-right-radius: 1000px;

border-top-right-radius: 1000px;

border-bottom-right-radius: 1000px;

}

/\* We use this to get basic styling on all basic form elements \*/

input[type="text"],

input[type="password"],

input[type="date"],

input[type="datetime"],

input[type="datetime-local"],

input[type="month"],

input[type="week"],

input[type="email"],

input[type="number"],

input[type="search"],

input[type="tel"],

input[type="time"],

input[type="url"],

textarea {

background-color: white;

font-family: inherit;

border: 1px solid #cccccc;

-webkit-box-shadow: inset 0 1px 2px rgba(0, 0, 0, 0.1);

box-shadow: inset 0 1px 2px rgba(0, 0, 0, 0.1);

color: rgba(0, 0, 0, 0.75);

display: block;

font-size: 0.875em;

margin: 0 0 1em 0;

padding: 0.5em;

height: 2.3125em;

width: 100%;

-moz-box-sizing: border-box;

-webkit-box-sizing: border-box;

box-sizing: border-box;

-webkit-transition: -webkit-box-shadow 0.45s, border-color 0.45s ease-in-out;

-moz-transition: -moz-box-shadow 0.45s, border-color 0.45s ease-in-out;

transition: box-shadow 0.45s, border-color 0.45s ease-in-out;

}

input[type="text"]:focus,

input[type="password"]:focus,

input[type="date"]:focus,

input[type="datetime"]:focus,

input[type="datetime-local"]:focus,

input[type="month"]:focus,

input[type="week"]:focus,

input[type="email"]:focus,

input[type="number"]:focus,

input[type="search"]:focus,

input[type="tel"]:focus,

input[type="time"]:focus,

input[type="url"]:focus,

textarea:focus {

-webkit-box-shadow: 0 0 5px #999999;

-moz-box-shadow: 0 0 5px #999999;

box-shadow: 0 0 5px #999999;

border-color: #999999;

}

input[type="text"]:focus,

input[type="password"]:focus,

input[type="date"]:focus,

input[type="datetime"]:focus,

input[type="datetime-local"]:focus,

input[type="month"]:focus,

input[type="week"]:focus,

input[type="email"]:focus,

input[type="number"]:focus,

input[type="search"]:focus,

input[type="tel"]:focus,

input[type="time"]:focus,

input[type="url"]:focus,

textarea:focus {

background: #fafafa;

border-color: #999999;

outline: none;

}

input[type="text"][disabled],

input[type="password"][disabled],

input[type="date"][disabled],

input[type="datetime"][disabled],

input[type="datetime-local"][disabled],

input[type="month"][disabled],

input[type="week"][disabled],

input[type="email"][disabled],

input[type="number"][disabled],

input[type="search"][disabled],

input[type="tel"][disabled],

input[type="time"][disabled],

input[type="url"][disabled],

textarea[disabled] {

background-color: #dddddd;

}

/\* Adjust margin for form elements below \*/

input[type="file"],

input[type="checkbox"],

input[type="radio"],

select {

margin: 8px 40px 1em;

}

/\* Normalize file input width \*/

input[type="file"] {

width: 100%;

}

/\* We add basic fieldset styling \*/

fieldset {

border: solid 1px #dddddd;

padding: 1.25em;

margin: 1.125em 0;

}

fieldset legend {

font-weight: bold;

background: white;

padding: 0 0.1875em;

margin: 0;

margin-left: -0.1875em;

}

/\* Error Handling \*/

[data-abide] .error small.error, [data-abide] span.error, [data-abide] small.error {

display: block;

padding: 0.375em 0.25em;

margin-top: -1.3125em;

margin-bottom: 1em;

font-size: 0.75em;

font-weight: bold;

background: #c60f13;

color: white;

margin-top: 0;

}

[data-abide] span.error, [data-abide] small.error {

display: none;

}

span.error, small.error {

display: block;

padding: 0.375em 0.25em;

margin-top: -1.3125em;

margin-bottom: 1em;

font-size: 0.75em;

font-weight: bold;

background: #c60f13;

color: white;

}

.error input,

.error textarea,

.error select {

border-color: #c60f13;

background-color: rgba(198, 15, 19, 0.1);

margin-bottom: 0;

}

.error input:focus,

.error textarea:focus,

.error select:focus {

background: #fafafa;

border-color: #999999;

}

.error label,

.error label.error {

color: #c60f13;

}

.error small.error {

display: block;

padding: 0.375em 0.25em;

margin-top: -1.3125em;

margin-bottom: 1em;

font-size: 0.75em;

font-weight: bold;

background: #c60f13;

color: white;

}

.error span.error-message {

display: block;

}

input.error,

textarea.error {

border-color: #c60f13;

background-color: rgba(198, 15, 19, 0.1);

}

input.error:focus,

textarea.error:focus {

background: #fafafa;

border-color: #999999;

}

.error select {

border-color: #c60f13;

background-color: rgba(198, 15, 19, 0.1);

}

.error select:focus {

background: #fafafa;

border-color: #999999;

}

label.error {

color: #c60f13;

}

/\* Button Groups \*/

.button-group {

list-style: none;

margin: 0;

\*zoom: 1;

}

.button-group:before, .button-group:after {

content: " ";

display: table;

}

.button-group:after {

clear: both;

}

.button-group > \* {

margin: 0 0 0 -1px;

float: left;

}

.button-group > \*:first-child {

margin-left: 0;

}

.button-group.radius > \*:first-child, .button-group.radius > \*:first-child > a, .button-group.radius > \*:first-child > button, .button-group.radius > \*:first-child > .button {

-moz-border-radius-bottomleft: 3px;

-moz-border-radius-topleft: 3px;

-webkit-border-bottom-left-radius: 3px;

-webkit-border-top-left-radius: 3px;

border-bottom-left-radius: 3px;

border-top-left-radius: 3px;

}

.button-group.radius > \*:last-child, .button-group.radius > \*:last-child > a, .button-group.radius > \*:last-child > button, .button-group.radius > \*:last-child > .button {

-moz-border-radius-topright: 3px;

-moz-border-radius-bottomright: 3px;

-webkit-border-top-right-radius: 3px;

-webkit-border-bottom-right-radius: 3px;

border-top-right-radius: 3px;

border-bottom-right-radius: 3px;

}

.button-group.round > \*:first-child, .button-group.round > \*:first-child > a, .button-group.round > \*:first-child > button, .button-group.round > \*:first-child > .button {

-moz-border-radius-bottomleft: 1000px;

-moz-border-radius-topleft: 1000px;

-webkit-border-bottom-left-radius: 1000px;

-webkit-border-top-left-radius: 1000px;

border-bottom-left-radius: 1000px;

border-top-left-radius: 1000px;

}

.button-group.round > \*:last-child, .button-group.round > \*:last-child > a, .button-group.round > \*:last-child > button, .button-group.round > \*:last-child > .button {

-moz-border-radius-topright: 1000px;

-moz-border-radius-bottomright: 1000px;

-webkit-border-top-right-radius: 1000px;

-webkit-border-bottom-right-radius: 1000px;

border-top-right-radius: 1000px;

border-bottom-right-radius: 1000px;

}

.button-group.even-2 li {

width: 50%;

}

.button-group.even-2 li button, .button-group.even-2 li .button {

width: 100%;

}

.button-group.even-3 li {

width: 33.33333%;

}

.button-group.even-3 li button, .button-group.even-3 li .button {

width: 100%;

}

.button-group.even-4 li {

width: 25%;

}

.button-group.even-4 li button, .button-group.even-4 li .button {

width: 100%;

}

.button-group.even-5 li {

width: 20%;

}

.button-group.even-5 li button, .button-group.even-5 li .button {

width: 100%;

}

.button-group.even-6 li {

width: 16.66667%;

}

.button-group.even-6 li button, .button-group.even-6 li .button {

width: 100%;

}

.button-group.even-7 li {

width: 14.28571%;

}

.button-group.even-7 li button, .button-group.even-7 li .button {

width: 100%;

}

.button-group.even-8 li {

width: 12.5%;

}

.button-group.even-8 li button, .button-group.even-8 li .button {

width: 100%;

}

.button-bar {

\*zoom: 1;

}

.button-bar:before, .button-bar:after {

content: " ";

display: table;

}

.button-bar:after {

clear: both;

}

.button-bar .button-group {

float: left;

margin-right: 0.625em;

}

.button-bar .button-group div {

overflow: hidden;

}

/\* Dropdown Button \*/

.dropdown.button {

position: relative;

padding-right: 3.1875em;

}

.dropdown.button:before {

position: absolute;

content: "";

width: 0;

height: 0;

display: block;

border-style: solid;

border-color: white transparent transparent transparent;

top: 50%;

}

.dropdown.button:before {

border-width: 0.5625em;

right: 1.5em;

margin-top: -0.25em;

}

.dropdown.button:before {

border-color: white transparent transparent transparent;

}

.dropdown.button.tiny {

padding-right: 2.1875em;

}

.dropdown.button.tiny:before {

border-width: 0.4375em;

right: 0.875em;

margin-top: -0.15625em;

}

.dropdown.button.tiny:before {

border-color: white transparent transparent transparent;

}

.dropdown.button.small {

padding-right: 2.8125em;

}

.dropdown.button.small:before {

border-width: 0.5625em;

right: 1.125em;

margin-top: -0.21875em;

}

.dropdown.button.small:before {

border-color: white transparent transparent transparent;

}

.dropdown.button.large {

padding-right: 4em;

}

.dropdown.button.large:before {

border-width: 0.625em;

right: 1.75em;

margin-top: -0.3125em;

}

.dropdown.button.large:before {

border-color: white transparent transparent transparent;

}

.dropdown.button.secondary:before {

border-color: #333333 transparent transparent transparent;

}

/\* Split Buttons \*/

.split.button {

position: relative;

padding-right: 4.8em;

}

.split.button span {

display: block;

height: 100%;

position: absolute;

right: 0;

top: 0;

border-left: solid 1px;

}

.split.button span:before {

position: absolute;

content: "";

width: 0;

height: 0;

display: block;

border-style: inset;

left: 50%;

}

.split.button span:active {

background-color: rgba(0, 0, 0, 0.1);

}

.split.button span {

border-left-color: #1e728c;

}

.split.button span {

width: 3em;

}

.split.button span:before {

border-top-style: solid;

border-width: 0.5625em;

top: 1.125em;

margin-left: -0.5625em;

}

.split.button span:before {

border-color: white transparent transparent transparent;

}

.split.button.secondary span {

border-left-color: #c3c3c3;

}

.split.button.secondary span:before {

border-color: white transparent transparent transparent;

}

.split.button.alert span {

border-left-color: #7f0a0c;

}

.split.button.success span {

border-left-color: #396516;

}

.split.button.tiny {

padding-right: 3.9375em;

}

.split.button.tiny span {

width: 2.84375em;

}

.split.button.tiny span:before {

border-top-style: solid;

border-width: 0.4375em;

top: 0.875em;

margin-left: -0.3125em;

}

.split.button.small {

padding-right: 3.9375em;

}

.split.button.small span {

width: 2.8125em;

}

.split.button.small span:before {

border-top-style: solid;

border-width: 0.5625em;

top: 0.84375em;

margin-left: -0.5625em;

}

.split.button.large {

padding-right: 6em;

}

.split.button.large span {

width: 3.75em;

}

.split.button.large span:before {

border-top-style: solid;

border-width: 0.625em;

top: 1.3125em;

margin-left: -0.5625em;

}

.split.button.expand {

padding-left: 2em;

}

.split.button.secondary span:before {

border-color: #333333 transparent transparent transparent;

}

.split.button.radius span {

-moz-border-radius-topright: 3px;

-moz-border-radius-bottomright: 3px;

-webkit-border-top-right-radius: 3px;

-webkit-border-bottom-right-radius: 3px;

border-top-right-radius: 3px;

border-bottom-right-radius: 3px;

}

.split.button.round span {

-moz-border-radius-topright: 1000px;

-moz-border-radius-bottomright: 1000px;

-webkit-border-top-right-radius: 1000px;

-webkit-border-bottom-right-radius: 1000px;

border-top-right-radius: 1000px;

border-bottom-right-radius: 1000px;

}

/\* Flex Video \*/

.flex-video {

position: relative;

padding-top: 1.5625em;

padding-bottom: 67.5%;

height: 0;

margin-bottom: 1em;

overflow: hidden;

}

.flex-video.widescreen {

padding-bottom: 57.25%;

}

.flex-video.vimeo {

padding-top: 0;

}

.flex-video iframe,

.flex-video object,

.flex-video embed,

.flex-video video {

position: absolute;

top: 0;

left: 0;

width: 100%;

height: 100%;

}

/\* Sections \*/

[data-section=''], [data-section='auto'], .section-container.auto,

[data-section='vertical-tabs'], .section-container.vertical-tabs,

[data-section='vertical-nav'], .section-container.vertical-nav,

[data-section='horizontal-nav'], .section-container.horizontal-nav,

[data-section='accordion'], .section-container.accordion {

width: 100%;

position: relative;

display: block;

margin-bottom: 1.25em;

}

[data-section=''][data-section-small-style], [data-section='auto'][data-section-small-style], .section-container.auto[data-section-small-style],

[data-section='vertical-tabs'][data-section-small-style], .section-container.vertical-tabs[data-section-small-style],

[data-section='vertical-nav'][data-section-small-style], .section-container.vertical-nav[data-section-small-style],

[data-section='horizontal-nav'][data-section-small-style], .section-container.horizontal-nav[data-section-small-style],

[data-section='accordion'][data-section-small-style], .section-container.accordion[data-section-small-style] {

width: 100% !important;

}

[data-section=''][data-section-small-style] > [data-section-region], [data-section=''][data-section-small-style] > section, [data-section=''][data-section-small-style] > .section, [data-section='auto'][data-section-small-style] > [data-section-region], [data-section='auto'][data-section-small-style] > section, [data-section='auto'][data-section-small-style] > .section, .section-container.auto[data-section-small-style] > [data-section-region], .section-container.auto[data-section-small-style] > section, .section-container.auto[data-section-small-style] > .section,

[data-section='vertical-tabs'][data-section-small-style] > [data-section-region],

[data-section='vertical-tabs'][data-section-small-style] > section,

[data-section='vertical-tabs'][data-section-small-style] > .section, .section-container.vertical-tabs[data-section-small-style] > [data-section-region], .section-container.vertical-tabs[data-section-small-style] > section, .section-container.vertical-tabs[data-section-small-style] > .section,

[data-section='vertical-nav'][data-section-small-style] > [data-section-region],

[data-section='vertical-nav'][data-section-small-style] > section,

[data-section='vertical-nav'][data-section-small-style] > .section, .section-container.vertical-nav[data-section-small-style] > [data-section-region], .section-container.vertical-nav[data-section-small-style] > section, .section-container.vertical-nav[data-section-small-style] > .section,

[data-section='horizontal-nav'][data-section-small-style] > [data-section-region],

[data-section='horizontal-nav'][data-section-small-style] > section,

[data-section='horizontal-nav'][data-section-small-style] > .section, .section-container.horizontal-nav[data-section-small-style] > [data-section-region], .section-container.horizontal-nav[data-section-small-style] > section, .section-container.horizontal-nav[data-section-small-style] > .section,

[data-section='accordion'][data-section-small-style] > [data-section-region],

[data-section='accordion'][data-section-small-style] > section,

[data-section='accordion'][data-section-small-style] > .section, .section-container.accordion[data-section-small-style] > [data-section-region], .section-container.accordion[data-section-small-style] > section, .section-container.accordion[data-section-small-style] > .section {

padding: 0 !important;

margin: 0 !important;

}

[data-section=''][data-section-small-style] > [data-section-region] > [data-section-title], [data-section=''][data-section-small-style] > [data-section-region] > .title, [data-section=''][data-section-small-style] > section > [data-section-title], [data-section=''][data-section-small-style] > section > .title, [data-section=''][data-section-small-style] > .section > [data-section-title], [data-section=''][data-section-small-style] > .section > .title, [data-section='auto'][data-section-small-style] > [data-section-region] > [data-section-title], [data-section='auto'][data-section-small-style] > [data-section-region] > .title, [data-section='auto'][data-section-small-style] > section > [data-section-title], [data-section='auto'][data-section-small-style] > section > .title, [data-section='auto'][data-section-small-style] > .section > [data-section-title], [data-section='auto'][data-section-small-style] > .section > .title, .section-container.auto[data-section-small-style] > [data-section-region] > [data-section-title], .section-container.auto[data-section-small-style] > [data-section-region] > .title, .section-container.auto[data-section-small-style] > section > [data-section-title], .section-container.auto[data-section-small-style] > section > .title, .section-container.auto[data-section-small-style] > .section > [data-section-title], .section-container.auto[data-section-small-style] > .section > .title,

[data-section='vertical-tabs'][data-section-small-style] > [data-section-region] > [data-section-title],

[data-section='vertical-tabs'][data-section-small-style] > [data-section-region] > .title,

[data-section='vertical-tabs'][data-section-small-style] > section > [data-section-title],

[data-section='vertical-tabs'][data-section-small-style] > section > .title,

[data-section='vertical-tabs'][data-section-small-style] > .section > [data-section-title],

[data-section='vertical-tabs'][data-section-small-style] > .section > .title, .section-container.vertical-tabs[data-section-small-style] > [data-section-region] > [data-section-title], .section-container.vertical-tabs[data-section-small-style] > [data-section-region] > .title, .section-container.vertical-tabs[data-section-small-style] > section > [data-section-title], .section-container.vertical-tabs[data-section-small-style] > section > .title, .section-container.vertical-tabs[data-section-small-style] > .section > [data-section-title], .section-container.vertical-tabs[data-section-small-style] > .section > .title,

[data-section='vertical-nav'][data-section-small-style] > [data-section-region] > [data-section-title],

[data-section='vertical-nav'][data-section-small-style] > [data-section-region] > .title,

[data-section='vertical-nav'][data-section-small-style] > section > [data-section-title],

[data-section='vertical-nav'][data-section-small-style] > section > .title,

[data-section='vertical-nav'][data-section-small-style] > .section > [data-section-title],

[data-section='vertical-nav'][data-section-small-style] > .section > .title, .section-container.vertical-nav[data-section-small-style] > [data-section-region] > [data-section-title], .section-container.vertical-nav[data-section-small-style] > [data-section-region] > .title, .section-container.vertical-nav[data-section-small-style] > section > [data-section-title], .section-container.vertical-nav[data-section-small-style] > section > .title, .section-container.vertical-nav[data-section-small-style] > .section > [data-section-title], .section-container.vertical-nav[data-section-small-style] > .section > .title,

[data-section='horizontal-nav'][data-section-small-style] > [data-section-region] > [data-section-title],

[data-section='horizontal-nav'][data-section-small-style] > [data-section-region] > .title,

[data-section='horizontal-nav'][data-section-small-style] > section > [data-section-title],

[data-section='horizontal-nav'][data-section-small-style] > section > .title,

[data-section='horizontal-nav'][data-section-small-style] > .section > [data-section-title],

[data-section='horizontal-nav'][data-section-small-style] > .section > .title, .section-container.horizontal-nav[data-section-small-style] > [data-section-region] > [data-section-title], .section-container.horizontal-nav[data-section-small-style] > [data-section-region] > .title, .section-container.horizontal-nav[data-section-small-style] > section > [data-section-title], .section-container.horizontal-nav[data-section-small-style] > section > .title, .section-container.horizontal-nav[data-section-small-style] > .section > [data-section-title], .section-container.horizontal-nav[data-section-small-style] > .section > .title,

[data-section='accordion'][data-section-small-style] > [data-section-region] > [data-section-title],

[data-section='accordion'][data-section-small-style] > [data-section-region] > .title,

[data-section='accordion'][data-section-small-style] > section > [data-section-title],

[data-section='accordion'][data-section-small-style] > section > .title,

[data-section='accordion'][data-section-small-style] > .section > [data-section-title],

[data-section='accordion'][data-section-small-style] > .section > .title, .section-container.accordion[data-section-small-style] > [data-section-region] > [data-section-title], .section-container.accordion[data-section-small-style] > [data-section-region] > .title, .section-container.accordion[data-section-small-style] > section > [data-section-title], .section-container.accordion[data-section-small-style] > section > .title, .section-container.accordion[data-section-small-style] > .section > [data-section-title], .section-container.accordion[data-section-small-style] > .section > .title {

width: 100% !important;

}

[data-section=''] > section, [data-section=''] > .section, [data-section=''] > [data-section-region], [data-section='auto'] > section, [data-section='auto'] > .section, [data-section='auto'] > [data-section-region], .section-container.auto > section, .section-container.auto > .section, .section-container.auto > [data-section-region],

[data-section='vertical-tabs'] > section,

[data-section='vertical-tabs'] > .section,

[data-section='vertical-tabs'] > [data-section-region], .section-container.vertical-tabs > section, .section-container.vertical-tabs > .section, .section-container.vertical-tabs > [data-section-region],

[data-section='vertical-nav'] > section,

[data-section='vertical-nav'] > .section,

[data-section='vertical-nav'] > [data-section-region], .section-container.vertical-nav > section, .section-container.vertical-nav > .section, .section-container.vertical-nav > [data-section-region],

[data-section='horizontal-nav'] > section,

[data-section='horizontal-nav'] > .section,

[data-section='horizontal-nav'] > [data-section-region], .section-container.horizontal-nav > section, .section-container.horizontal-nav > .section, .section-container.horizontal-nav > [data-section-region],

[data-section='accordion'] > section,

[data-section='accordion'] > .section,

[data-section='accordion'] > [data-section-region], .section-container.accordion > section, .section-container.accordion > .section, .section-container.accordion > [data-section-region] {

margin: 0;

}

[data-section=''] > section > [data-section-title], [data-section=''] > section > .title, [data-section=''] > .section > [data-section-title], [data-section=''] > .section > .title, [data-section=''] > [data-section-region] > [data-section-title], [data-section=''] > [data-section-region] > .title, [data-section='auto'] > section > [data-section-title], [data-section='auto'] > section > .title, [data-section='auto'] > .section > [data-section-title], [data-section='auto'] > .section > .title, [data-section='auto'] > [data-section-region] > [data-section-title], [data-section='auto'] > [data-section-region] > .title, .section-container.auto > section > [data-section-title], .section-container.auto > section > .title, .section-container.auto > .section > [data-section-title], .section-container.auto > .section > .title, .section-container.auto > [data-section-region] > [data-section-title], .section-container.auto > [data-section-region] > .title,

[data-section='vertical-tabs'] > section > [data-section-title],

[data-section='vertical-tabs'] > section > .title,

[data-section='vertical-tabs'] > .section > [data-section-title],

[data-section='vertical-tabs'] > .section > .title,

[data-section='vertical-tabs'] > [data-section-region] > [data-section-title],

[data-section='vertical-tabs'] > [data-section-region] > .title, .section-container.vertical-tabs > section > [data-section-title], .section-container.vertical-tabs > section > .title, .section-container.vertical-tabs > .section > [data-section-title], .section-container.vertical-tabs > .section > .title, .section-container.vertical-tabs > [data-section-region] > [data-section-title], .section-container.vertical-tabs > [data-section-region] > .title,

[data-section='vertical-nav'] > section > [data-section-title],

[data-section='vertical-nav'] > section > .title,

[data-section='vertical-nav'] > .section > [data-section-title],

[data-section='vertical-nav'] > .section > .title,

[data-section='vertical-nav'] > [data-section-region] > [data-section-title],

[data-section='vertical-nav'] > [data-section-region] > .title, .section-container.vertical-nav > section > [data-section-title], .section-container.vertical-nav > section > .title, .section-container.vertical-nav > .section > [data-section-title], .section-container.vertical-nav > .section > .title, .section-container.vertical-nav > [data-section-region] > [data-section-title], .section-container.vertical-nav > [data-section-region] > .title,

[data-section='horizontal-nav'] > section > [data-section-title],

[data-section='horizontal-nav'] > section > .title,

[data-section='horizontal-nav'] > .section > [data-section-title],

[data-section='horizontal-nav'] > .section > .title,

[data-section='horizontal-nav'] > [data-section-region] > [data-section-title],

[data-section='horizontal-nav'] > [data-section-region] > .title, .section-container.horizontal-nav > section > [data-section-title], .section-container.horizontal-nav > section > .title, .section-container.horizontal-nav > .section > [data-section-title], .section-container.horizontal-nav > .section > .title, .section-container.horizontal-nav > [data-section-region] > [data-section-title], .section-container.horizontal-nav > [data-section-region] > .title,

[data-section='accordion'] > section > [data-section-title],

[data-section='accordion'] > section > .title,

[data-section='accordion'] > .section > [data-section-title],

[data-section='accordion'] > .section > .title,

[data-section='accordion'] > [data-section-region] > [data-section-title],

[data-section='accordion'] > [data-section-region] > .title, .section-container.accordion > section > [data-section-title], .section-container.accordion > section > .title, .section-container.accordion > .section > [data-section-title], .section-container.accordion > .section > .title, .section-container.accordion > [data-section-region] > [data-section-title], .section-container.accordion > [data-section-region] > .title {

margin-bottom: 0;

}

[data-section=''] > section > [data-section-title] a, [data-section=''] > section > .title a, [data-section=''] > .section > [data-section-title] a, [data-section=''] > .section > .title a, [data-section=''] > [data-section-region] > [data-section-title] a, [data-section=''] > [data-section-region] > .title a, [data-section='auto'] > section > [data-section-title] a, [data-section='auto'] > section > .title a, [data-section='auto'] > .section > [data-section-title] a, [data-section='auto'] > .section > .title a, [data-section='auto'] > [data-section-region] > [data-section-title] a, [data-section='auto'] > [data-section-region] > .title a, .section-container.auto > section > [data-section-title] a, .section-container.auto > section > .title a, .section-container.auto > .section > [data-section-title] a, .section-container.auto > .section > .title a, .section-container.auto > [data-section-region] > [data-section-title] a, .section-container.auto > [data-section-region] > .title a,

[data-section='vertical-tabs'] > section > [data-section-title] a,

[data-section='vertical-tabs'] > section > .title a,

[data-section='vertical-tabs'] > .section > [data-section-title] a,

[data-section='vertical-tabs'] > .section > .title a,

[data-section='vertical-tabs'] > [data-section-region] > [data-section-title] a,

[data-section='vertical-tabs'] > [data-section-region] > .title a, .section-container.vertical-tabs > section > [data-section-title] a, .section-container.vertical-tabs > section > .title a, .section-container.vertical-tabs > .section > [data-section-title] a, .section-container.vertical-tabs > .section > .title a, .section-container.vertical-tabs > [data-section-region] > [data-section-title] a, .section-container.vertical-tabs > [data-section-region] > .title a,

[data-section='vertical-nav'] > section > [data-section-title] a,

[data-section='vertical-nav'] > section > .title a,

[data-section='vertical-nav'] > .section > [data-section-title] a,

[data-section='vertical-nav'] > .section > .title a,

[data-section='vertical-nav'] > [data-section-region] > [data-section-title] a,

[data-section='vertical-nav'] > [data-section-region] > .title a, .section-container.vertical-nav > section > [data-section-title] a, .section-container.vertical-nav > section > .title a, .section-container.vertical-nav > .section > [data-section-title] a, .section-container.vertical-nav > .section > .title a, .section-container.vertical-nav > [data-section-region] > [data-section-title] a, .section-container.vertical-nav > [data-section-region] > .title a,

[data-section='horizontal-nav'] > section > [data-section-title] a,

[data-section='horizontal-nav'] > section > .title a,

[data-section='horizontal-nav'] > .section > [data-section-title] a,

[data-section='horizontal-nav'] > .section > .title a,

[data-section='horizontal-nav'] > [data-section-region] > [data-section-title] a,

[data-section='horizontal-nav'] > [data-section-region] > .title a, .section-container.horizontal-nav > section > [data-section-title] a, .section-container.horizontal-nav > section > .title a, .section-container.horizontal-nav > .section > [data-section-title] a, .section-container.horizontal-nav > .section > .title a, .section-container.horizontal-nav > [data-section-region] > [data-section-title] a, .section-container.horizontal-nav > [data-section-region] > .title a,

[data-section='accordion'] > section > [data-section-title] a,

[data-section='accordion'] > section > .title a,

[data-section='accordion'] > .section > [data-section-title] a,

[data-section='accordion'] > .section > .title a,

[data-section='accordion'] > [data-section-region] > [data-section-title] a,

[data-section='accordion'] > [data-section-region] > .title a, .section-container.accordion > section > [data-section-title] a, .section-container.accordion > section > .title a, .section-container.accordion > .section > [data-section-title] a, .section-container.accordion > .section > .title a, .section-container.accordion > [data-section-region] > [data-section-title] a, .section-container.accordion > [data-section-region] > .title a {

width: 100%;

display: inline-block;

white-space: nowrap;

}

[data-section=''] > section > [data-section-content], [data-section=''] > section > .content, [data-section=''] > .section > [data-section-content], [data-section=''] > .section > .content, [data-section=''] > [data-section-region] > [data-section-content], [data-section=''] > [data-section-region] > .content, [data-section='auto'] > section > [data-section-content], [data-section='auto'] > section > .content, [data-section='auto'] > .section > [data-section-content], [data-section='auto'] > .section > .content, [data-section='auto'] > [data-section-region] > [data-section-content], [data-section='auto'] > [data-section-region] > .content, .section-container.auto > section > [data-section-content], .section-container.auto > section > .content, .section-container.auto > .section > [data-section-content], .section-container.auto > .section > .content, .section-container.auto > [data-section-region] > [data-section-content], .section-container.auto > [data-section-region] > .content,

[data-section='vertical-tabs'] > section > [data-section-content],

[data-section='vertical-tabs'] > section > .content,

[data-section='vertical-tabs'] > .section > [data-section-content],

[data-section='vertical-tabs'] > .section > .content,

[data-section='vertical-tabs'] > [data-section-region] > [data-section-content],

[data-section='vertical-tabs'] > [data-section-region] > .content, .section-container.vertical-tabs > section > [data-section-content], .section-container.vertical-tabs > section > .content, .section-container.vertical-tabs > .section > [data-section-content], .section-container.vertical-tabs > .section > .content, .section-container.vertical-tabs > [data-section-region] > [data-section-content], .section-container.vertical-tabs > [data-section-region] > .content,

[data-section='vertical-nav'] > section > [data-section-content],

[data-section='vertical-nav'] > section > .content,

[data-section='vertical-nav'] > .section > [data-section-content],

[data-section='vertical-nav'] > .section > .content,

[data-section='vertical-nav'] > [data-section-region] > [data-section-content],

[data-section='vertical-nav'] > [data-section-region] > .content, .section-container.vertical-nav > section > [data-section-content], .section-container.vertical-nav > section > .content, .section-container.vertical-nav > .section > [data-section-content], .section-container.vertical-nav > .section > .content, .section-container.vertical-nav > [data-section-region] > [data-section-content], .section-container.vertical-nav > [data-section-region] > .content,

[data-section='horizontal-nav'] > section > [data-section-content],

[data-section='horizontal-nav'] > section > .content,

[data-section='horizontal-nav'] > .section > [data-section-content],

[data-section='horizontal-nav'] > .section > .content,

[data-section='horizontal-nav'] > [data-section-region] > [data-section-content],

[data-section='horizontal-nav'] > [data-section-region] > .content, .section-container.horizontal-nav > section > [data-section-content], .section-container.horizontal-nav > section > .content, .section-container.horizontal-nav > .section > [data-section-content], .section-container.horizontal-nav > .section > .content, .section-container.horizontal-nav > [data-section-region] > [data-section-content], .section-container.horizontal-nav > [data-section-region] > .content,

[data-section='accordion'] > section > [data-section-content],

[data-section='accordion'] > section > .content,

[data-section='accordion'] > .section > [data-section-content],

[data-section='accordion'] > .section > .content,

[data-section='accordion'] > [data-section-region] > [data-section-content],

[data-section='accordion'] > [data-section-region] > .content, .section-container.accordion > section > [data-section-content], .section-container.accordion > section > .content, .section-container.accordion > .section > [data-section-content], .section-container.accordion > .section > .content, .section-container.accordion > [data-section-region] > [data-section-content], .section-container.accordion > [data-section-region] > .content {

display: none;

}

[data-section=''] > section.active > [data-section-content], [data-section=''] > section.active > .content, [data-section=''] > .section.active > [data-section-content], [data-section=''] > .section.active > .content, [data-section=''] > [data-section-region].active > [data-section-content], [data-section=''] > [data-section-region].active > .content, [data-section='auto'] > section.active > [data-section-content], [data-section='auto'] > section.active > .content, [data-section='auto'] > .section.active > [data-section-content], [data-section='auto'] > .section.active > .content, [data-section='auto'] > [data-section-region].active > [data-section-content], [data-section='auto'] > [data-section-region].active > .content, .section-container.auto > section.active > [data-section-content], .section-container.auto > section.active > .content, .section-container.auto > .section.active > [data-section-content], .section-container.auto > .section.active > .content, .section-container.auto > [data-section-region].active > [data-section-content], .section-container.auto > [data-section-region].active > .content,

[data-section='vertical-tabs'] > section.active > [data-section-content],

[data-section='vertical-tabs'] > section.active > .content,

[data-section='vertical-tabs'] > .section.active > [data-section-content],

[data-section='vertical-tabs'] > .section.active > .content,

[data-section='vertical-tabs'] > [data-section-region].active > [data-section-content],

[data-section='vertical-tabs'] > [data-section-region].active > .content, .section-container.vertical-tabs > section.active > [data-section-content], .section-container.vertical-tabs > section.active > .content, .section-container.vertical-tabs > .section.active > [data-section-content], .section-container.vertical-tabs > .section.active > .content, .section-container.vertical-tabs > [data-section-region].active > [data-section-content], .section-container.vertical-tabs > [data-section-region].active > .content,

[data-section='vertical-nav'] > section.active > [data-section-content],

[data-section='vertical-nav'] > section.active > .content,

[data-section='vertical-nav'] > .section.active > [data-section-content],

[data-section='vertical-nav'] > .section.active > .content,

[data-section='vertical-nav'] > [data-section-region].active > [data-section-content],

[data-section='vertical-nav'] > [data-section-region].active > .content, .section-container.vertical-nav > section.active > [data-section-content], .section-container.vertical-nav > section.active > .content, .section-container.vertical-nav > .section.active > [data-section-content], .section-container.vertical-nav > .section.active > .content, .section-container.vertical-nav > [data-section-region].active > [data-section-content], .section-container.vertical-nav > [data-section-region].active > .content,

[data-section='horizontal-nav'] > section.active > [data-section-content],

[data-section='horizontal-nav'] > section.active > .content,

[data-section='horizontal-nav'] > .section.active > [data-section-content],

[data-section='horizontal-nav'] > .section.active > .content,

[data-section='horizontal-nav'] > [data-section-region].active > [data-section-content],

[data-section='horizontal-nav'] > [data-section-region].active > .content, .section-container.horizontal-nav > section.active > [data-section-content], .section-container.horizontal-nav > section.active > .content, .section-container.horizontal-nav > .section.active > [data-section-content], .section-container.horizontal-nav > .section.active > .content, .section-container.horizontal-nav > [data-section-region].active > [data-section-content], .section-container.horizontal-nav > [data-section-region].active > .content,

[data-section='accordion'] > section.active > [data-section-content],

[data-section='accordion'] > section.active > .content,

[data-section='accordion'] > .section.active > [data-section-content],

[data-section='accordion'] > .section.active > .content,

[data-section='accordion'] > [data-section-region].active > [data-section-content],

[data-section='accordion'] > [data-section-region].active > .content, .section-container.accordion > section.active > [data-section-content], .section-container.accordion > section.active > .content, .section-container.accordion > .section.active > [data-section-content], .section-container.accordion > .section.active > .content, .section-container.accordion > [data-section-region].active > [data-section-content], .section-container.accordion > [data-section-region].active > .content {

display: block;

}

[data-section=''] > section:not(.active), [data-section=''] > .section:not(.active), [data-section=''] > [data-section-region]:not(.active), [data-section='auto'] > section:not(.active), [data-section='auto'] > .section:not(.active), [data-section='auto'] > [data-section-region]:not(.active), .section-container.auto > section:not(.active), .section-container.auto > .section:not(.active), .section-container.auto > [data-section-region]:not(.active),

[data-section='vertical-tabs'] > section:not(.active),

[data-section='vertical-tabs'] > .section:not(.active),

[data-section='vertical-tabs'] > [data-section-region]:not(.active), .section-container.vertical-tabs > section:not(.active), .section-container.vertical-tabs > .section:not(.active), .section-container.vertical-tabs > [data-section-region]:not(.active),

[data-section='vertical-nav'] > section:not(.active),

[data-section='vertical-nav'] > .section:not(.active),

[data-section='vertical-nav'] > [data-section-region]:not(.active), .section-container.vertical-nav > section:not(.active), .section-container.vertical-nav > .section:not(.active), .section-container.vertical-nav > [data-section-region]:not(.active),

[data-section='horizontal-nav'] > section:not(.active),

[data-section='horizontal-nav'] > .section:not(.active),

[data-section='horizontal-nav'] > [data-section-region]:not(.active), .section-container.horizontal-nav > section:not(.active), .section-container.horizontal-nav > .section:not(.active), .section-container.horizontal-nav > [data-section-region]:not(.active),

[data-section='accordion'] > section:not(.active),

[data-section='accordion'] > .section:not(.active),

[data-section='accordion'] > [data-section-region]:not(.active), .section-container.accordion > section:not(.active), .section-container.accordion > .section:not(.active), .section-container.accordion > [data-section-region]:not(.active) {

padding: 0 !important;

}

[data-section=''] > section > [data-section-title], [data-section=''] > section > .title, [data-section=''] > .section > [data-section-title], [data-section=''] > .section > .title, [data-section=''] > [data-section-region] > [data-section-title], [data-section=''] > [data-section-region] > .title, [data-section='auto'] > section > [data-section-title], [data-section='auto'] > section > .title, [data-section='auto'] > .section > [data-section-title], [data-section='auto'] > .section > .title, [data-section='auto'] > [data-section-region] > [data-section-title], [data-section='auto'] > [data-section-region] > .title, .section-container.auto > section > [data-section-title], .section-container.auto > section > .title, .section-container.auto > .section > [data-section-title], .section-container.auto > .section > .title, .section-container.auto > [data-section-region] > [data-section-title], .section-container.auto > [data-section-region] > .title,

[data-section='vertical-tabs'] > section > [data-section-title],

[data-section='vertical-tabs'] > section > .title,

[data-section='vertical-tabs'] > .section > [data-section-title],

[data-section='vertical-tabs'] > .section > .title,

[data-section='vertical-tabs'] > [data-section-region] > [data-section-title],

[data-section='vertical-tabs'] > [data-section-region] > .title, .section-container.vertical-tabs > section > [data-section-title], .section-container.vertical-tabs > section > .title, .section-container.vertical-tabs > .section > [data-section-title], .section-container.vertical-tabs > .section > .title, .section-container.vertical-tabs > [data-section-region] > [data-section-title], .section-container.vertical-tabs > [data-section-region] > .title,

[data-section='vertical-nav'] > section > [data-section-title],

[data-section='vertical-nav'] > section > .title,

[data-section='vertical-nav'] > .section > [data-section-title],

[data-section='vertical-nav'] > .section > .title,

[data-section='vertical-nav'] > [data-section-region] > [data-section-title],

[data-section='vertical-nav'] > [data-section-region] > .title, .section-container.vertical-nav > section > [data-section-title], .section-container.vertical-nav > section > .title, .section-container.vertical-nav > .section > [data-section-title], .section-container.vertical-nav > .section > .title, .section-container.vertical-nav > [data-section-region] > [data-section-title], .section-container.vertical-nav > [data-section-region] > .title,

[data-section='horizontal-nav'] > section > [data-section-title],

[data-section='horizontal-nav'] > section > .title,

[data-section='horizontal-nav'] > .section > [data-section-title],

[data-section='horizontal-nav'] > .section > .title,

[data-section='horizontal-nav'] > [data-section-region] > [data-section-title],

[data-section='horizontal-nav'] > [data-section-region] > .title, .section-container.horizontal-nav > section > [data-section-title], .section-container.horizontal-nav > section > .title, .section-container.horizontal-nav > .section > [data-section-title], .section-container.horizontal-nav > .section > .title, .section-container.horizontal-nav > [data-section-region] > [data-section-title], .section-container.horizontal-nav > [data-section-region] > .title,

[data-section='accordion'] > section > [data-section-title],

[data-section='accordion'] > section > .title,

[data-section='accordion'] > .section > [data-section-title],

[data-section='accordion'] > .section > .title,

[data-section='accordion'] > [data-section-region] > [data-section-title],

[data-section='accordion'] > [data-section-region] > .title, .section-container.accordion > section > [data-section-title], .section-container.accordion > section > .title, .section-container.accordion > .section > [data-section-title], .section-container.accordion > .section > .title, .section-container.accordion > [data-section-region] > [data-section-title], .section-container.accordion > [data-section-region] > .title {

width: 100%;

}

.section-container.auto,

.section-container.vertical-tabs,

.section-container.vertical-nav,

.section-container.horizontal-nav,

.section-container.accordion {

border-top: 1px solid #cccccc;

}

.section-container.auto > section > .title, .section-container.auto > .section > .title,

.section-container.vertical-tabs > section > .title,

.section-container.vertical-tabs > .section > .title,

.section-container.vertical-nav > section > .title,

.section-container.vertical-nav > .section > .title,

.section-container.horizontal-nav > section > .title,

.section-container.horizontal-nav > .section > .title,

.section-container.accordion > section > .title,

.section-container.accordion > .section > .title {

background-color: #efefef;

cursor: pointer;

border: solid 1px #cccccc;

}

.section-container.auto > section > .title a, .section-container.auto > .section > .title a,

.section-container.vertical-tabs > section > .title a,

.section-container.vertical-tabs > .section > .title a,

.section-container.vertical-nav > section > .title a,

.section-container.vertical-nav > .section > .title a,

.section-container.horizontal-nav > section > .title a,

.section-container.horizontal-nav > .section > .title a,

.section-container.accordion > section > .title a,

.section-container.accordion > .section > .title a {

padding: 0.9375em;

color: #333333;

font-size: 0.875em;

background: none;

}

.section-container.auto > section > .title:hover, .section-container.auto > .section > .title:hover,

.section-container.vertical-tabs > section > .title:hover,

.section-container.vertical-tabs > .section > .title:hover,

.section-container.vertical-nav > section > .title:hover,

.section-container.vertical-nav > .section > .title:hover,

.section-container.horizontal-nav > section > .title:hover,

.section-container.horizontal-nav > .section > .title:hover,

.section-container.accordion > section > .title:hover,

.section-container.accordion > .section > .title:hover {

background-color: #e2e2e2;

}

.section-container.auto > section > .content, .section-container.auto > .section > .content,

.section-container.vertical-tabs > section > .content,

.section-container.vertical-tabs > .section > .content,

.section-container.vertical-nav > section > .content,

.section-container.vertical-nav > .section > .content,

.section-container.horizontal-nav > section > .content,

.section-container.horizontal-nav > .section > .content,

.section-container.accordion > section > .content,

.section-container.accordion > .section > .content {

padding: 0.9375em;

background-color: white;

border: solid 1px #cccccc;

}

.section-container.auto > section > .content > \*:last-child, .section-container.auto > .section > .content > \*:last-child,

.section-container.vertical-tabs > section > .content > \*:last-child,

.section-container.vertical-tabs > .section > .content > \*:last-child,

.section-container.vertical-nav > section > .content > \*:last-child,

.section-container.vertical-nav > .section > .content > \*:last-child,

.section-container.horizontal-nav > section > .content > \*:last-child,

.section-container.horizontal-nav > .section > .content > \*:last-child,

.section-container.accordion > section > .content > \*:last-child,

.section-container.accordion > .section > .content > \*:last-child {

margin-bottom: 0;

}

.section-container.auto > section > .content > \*:first-child, .section-container.auto > .section > .content > \*:first-child,

.section-container.vertical-tabs > section > .content > \*:first-child,

.section-container.vertical-tabs > .section > .content > \*:first-child,

.section-container.vertical-nav > section > .content > \*:first-child,

.section-container.vertical-nav > .section > .content > \*:first-child,

.section-container.horizontal-nav > section > .content > \*:first-child,

.section-container.horizontal-nav > .section > .content > \*:first-child,

.section-container.accordion > section > .content > \*:first-child,

.section-container.accordion > .section > .content > \*:first-child {

padding-top: 0;

}

.section-container.auto > section > .content > \*:last-child:not(.flex-video), .section-container.auto > .section > .content > \*:last-child:not(.flex-video),

.section-container.vertical-tabs > section > .content > \*:last-child:not(.flex-video),

.section-container.vertical-tabs > .section > .content > \*:last-child:not(.flex-video),

.section-container.vertical-nav > section > .content > \*:last-child:not(.flex-video),

.section-container.vertical-nav > .section > .content > \*:last-child:not(.flex-video),

.section-container.horizontal-nav > section > .content > \*:last-child:not(.flex-video),

.section-container.horizontal-nav > .section > .content > \*:last-child:not(.flex-video),

.section-container.accordion > section > .content > \*:last-child:not(.flex-video),

.section-container.accordion > .section > .content > \*:last-child:not(.flex-video) {

padding-bottom: 0;

}

.section-container.auto > section.active > .title, .section-container.auto > .section.active > .title,

.section-container.vertical-tabs > section.active > .title,

.section-container.vertical-tabs > .section.active > .title,

.section-container.vertical-nav > section.active > .title,

.section-container.vertical-nav > .section.active > .title,

.section-container.horizontal-nav > section.active > .title,

.section-container.horizontal-nav > .section.active > .title,

.section-container.accordion > section.active > .title,

.section-container.accordion > .section.active > .title {

background: #d5d5d5;

}

.section-container.auto > section.active > .title a, .section-container.auto > .section.active > .title a,

.section-container.vertical-tabs > section.active > .title a,

.section-container.vertical-tabs > .section.active > .title a,

.section-container.vertical-nav > section.active > .title a,

.section-container.vertical-nav > .section.active > .title a,

.section-container.horizontal-nav > section.active > .title a,

.section-container.horizontal-nav > .section.active > .title a,

.section-container.accordion > section.active > .title a,

.section-container.accordion > .section.active > .title a {

color: #333333;

}

.section-container.auto > section:not(.active), .section-container.auto > .section:not(.active),

.section-container.vertical-tabs > section:not(.active),

.section-container.vertical-tabs > .section:not(.active),

.section-container.vertical-nav > section:not(.active),

.section-container.vertical-nav > .section:not(.active),

.section-container.horizontal-nav > section:not(.active),

.section-container.horizontal-nav > .section:not(.active),

.section-container.accordion > section:not(.active),

.section-container.accordion > .section:not(.active) {

padding: 0 !important;

}

.section-container.auto > section > .title, .section-container.auto > .section > .title,

.section-container.vertical-tabs > section > .title,

.section-container.vertical-tabs > .section > .title,

.section-container.vertical-nav > section > .title,

.section-container.vertical-nav > .section > .title,

.section-container.horizontal-nav > section > .title,

.section-container.horizontal-nav > .section > .title,

.section-container.accordion > section > .title,

.section-container.accordion > .section > .title {

border-top: none;

}

[data-section='tabs'], .section-container.tabs {

width: 100%;

position: relative;

display: block;

margin-bottom: 1.25em;

}

[data-section='tabs']:not([data-section-resized]):not([data-section-small-style]), .section-container.tabs:not([data-section-resized]):not([data-section-small-style]) {

visibility: hidden;

}

[data-section='tabs'] > section > [data-section-title], [data-section='tabs'] > section > .title, [data-section='tabs'] > .section > [data-section-title], [data-section='tabs'] > .section > .title, [data-section='tabs'] > [data-section-region] > [data-section-title], [data-section='tabs'] > [data-section-region] > .title, .section-container.tabs > section > [data-section-title], .section-container.tabs > section > .title, .section-container.tabs > .section > [data-section-title], .section-container.tabs > .section > .title, .section-container.tabs > [data-section-region] > [data-section-title], .section-container.tabs > [data-section-region] > .title {

margin-bottom: 0;

}

[data-section='tabs'] > section > [data-section-title] a, [data-section='tabs'] > section > .title a, [data-section='tabs'] > .section > [data-section-title] a, [data-section='tabs'] > .section > .title a, [data-section='tabs'] > [data-section-region] > [data-section-title] a, [data-section='tabs'] > [data-section-region] > .title a, .section-container.tabs > section > [data-section-title] a, .section-container.tabs > section > .title a, .section-container.tabs > .section > [data-section-title] a, .section-container.tabs > .section > .title a, .section-container.tabs > [data-section-region] > [data-section-title] a, .section-container.tabs > [data-section-region] > .title a {

width: 100%;

display: inline-block;

white-space: nowrap;

}

[data-section='tabs'] > section > [data-section-content], [data-section='tabs'] > section > .content, [data-section='tabs'] > .section > [data-section-content], [data-section='tabs'] > .section > .content, [data-section='tabs'] > [data-section-region] > [data-section-content], [data-section='tabs'] > [data-section-region] > .content, .section-container.tabs > section > [data-section-content], .section-container.tabs > section > .content, .section-container.tabs > .section > [data-section-content], .section-container.tabs > .section > .content, .section-container.tabs > [data-section-region] > [data-section-content], .section-container.tabs > [data-section-region] > .content {

display: none;

}

[data-section='tabs'] > section.active > [data-section-content], [data-section='tabs'] > section.active > .content, [data-section='tabs'] > .section.active > [data-section-content], [data-section='tabs'] > .section.active > .content, [data-section='tabs'] > [data-section-region].active > [data-section-content], [data-section='tabs'] > [data-section-region].active > .content, .section-container.tabs > section.active > [data-section-content], .section-container.tabs > section.active > .content, .section-container.tabs > .section.active > [data-section-content], .section-container.tabs > .section.active > .content, .section-container.tabs > [data-section-region].active > [data-section-content], .section-container.tabs > [data-section-region].active > .content {

display: block;

}

[data-section='tabs'] > section:not(.active), [data-section='tabs'] > .section:not(.active), [data-section='tabs'] > [data-section-region]:not(.active), .section-container.tabs > section:not(.active), .section-container.tabs > .section:not(.active), .section-container.tabs > [data-section-region]:not(.active) {

padding: 0 !important;

}

[data-section='tabs'] > section > [data-section-title], [data-section='tabs'] > section > .title, [data-section='tabs'] > .section > [data-section-title], [data-section='tabs'] > .section > .title, [data-section='tabs'] > [data-section-region] > [data-section-title], [data-section='tabs'] > [data-section-region] > .title, .section-container.tabs > section > [data-section-title], .section-container.tabs > section > .title, .section-container.tabs > .section > [data-section-title], .section-container.tabs > .section > .title, .section-container.tabs > [data-section-region] > [data-section-title], .section-container.tabs > [data-section-region] > .title {

width: auto;

position: absolute;

top: 0;

left: 0;

}

.section-container.tabs {

border: none;

}

.section-container.tabs > section > .title, .section-container.tabs > .section > .title {

background-color: #efefef;

cursor: pointer;

border: solid 1px #cccccc;

}

.section-container.tabs > section > .title a, .section-container.tabs > .section > .title a {

padding: 0.9375em;

color: #333333;

font-size: 0.875em;

background: none;

}

.section-container.tabs > section > .title:hover, .section-container.tabs > .section > .title:hover {

background-color: #e2e2e2;

}

.section-container.tabs > section > .content, .section-container.tabs > .section > .content {

padding: 0.9375em;

background-color: white;

border: solid 1px #cccccc;

}

.section-container.tabs > section > .content > \*:last-child, .section-container.tabs > .section > .content > \*:last-child {

margin-bottom: 0;

}

.section-container.tabs > section > .content > \*:first-child, .section-container.tabs > .section > .content > \*:first-child {

padding-top: 0;

}

.section-container.tabs > section > .content > \*:last-child:not(.flex-video), .section-container.tabs > .section > .content > \*:last-child:not(.flex-video) {

padding-bottom: 0;

}

.section-container.tabs > section.active > .title, .section-container.tabs > .section.active > .title {

background: white;

}

.section-container.tabs > section.active > .title a, .section-container.tabs > .section.active > .title a {

color: #333333;

}

.section-container.tabs > section:not(.active), .section-container.tabs > .section:not(.active) {

padding: 0 !important;

}

.section-container.tabs > section.active > .title, .section-container.tabs > .section.active > .title {

border-bottom: 0;

}

@media only screen and (min-width: 768px) {

[data-section=''], [data-section='auto'], .section-container.auto {

width: 100%;

position: relative;

display: block;

margin-bottom: 1.25em;

}

[data-section='']:not([data-section-resized]):not([data-section-small-style]), [data-section='auto']:not([data-section-resized]):not([data-section-small-style]), .section-container.auto:not([data-section-resized]):not([data-section-small-style]) {

visibility: hidden;

}

[data-section=''] > section > [data-section-title], [data-section=''] > section > .title, [data-section=''] > .section > [data-section-title], [data-section=''] > .section > .title, [data-section=''] > [data-section-region] > [data-section-title], [data-section=''] > [data-section-region] > .title, [data-section='auto'] > section > [data-section-title], [data-section='auto'] > section > .title, [data-section='auto'] > .section > [data-section-title], [data-section='auto'] > .section > .title, [data-section='auto'] > [data-section-region] > [data-section-title], [data-section='auto'] > [data-section-region] > .title, .section-container.auto > section > [data-section-title], .section-container.auto > section > .title, .section-container.auto > .section > [data-section-title], .section-container.auto > .section > .title, .section-container.auto > [data-section-region] > [data-section-title], .section-container.auto > [data-section-region] > .title {

margin-bottom: 0;

}

[data-section=''] > section > [data-section-title] a, [data-section=''] > section > .title a, [data-section=''] > .section > [data-section-title] a, [data-section=''] > .section > .title a, [data-section=''] > [data-section-region] > [data-section-title] a, [data-section=''] > [data-section-region] > .title a, [data-section='auto'] > section > [data-section-title] a, [data-section='auto'] > section > .title a, [data-section='auto'] > .section > [data-section-title] a, [data-section='auto'] > .section > .title a, [data-section='auto'] > [data-section-region] > [data-section-title] a, [data-section='auto'] > [data-section-region] > .title a, .section-container.auto > section > [data-section-title] a, .section-container.auto > section > .title a, .section-container.auto > .section > [data-section-title] a, .section-container.auto > .section > .title a, .section-container.auto > [data-section-region] > [data-section-title] a, .section-container.auto > [data-section-region] > .title a {

width: 100%;

display: inline-block;

white-space: nowrap;

}

[data-section=''] > section > [data-section-content], [data-section=''] > section > .content, [data-section=''] > .section > [data-section-content], [data-section=''] > .section > .content, [data-section=''] > [data-section-region] > [data-section-content], [data-section=''] > [data-section-region] > .content, [data-section='auto'] > section > [data-section-content], [data-section='auto'] > section > .content, [data-section='auto'] > .section > [data-section-content], [data-section='auto'] > .section > .content, [data-section='auto'] > [data-section-region] > [data-section-content], [data-section='auto'] > [data-section-region] > .content, .section-container.auto > section > [data-section-content], .section-container.auto > section > .content, .section-container.auto > .section > [data-section-content], .section-container.auto > .section > .content, .section-container.auto > [data-section-region] > [data-section-content], .section-container.auto > [data-section-region] > .content {

display: none;

}

[data-section=''] > section.active > [data-section-content], [data-section=''] > section.active > .content, [data-section=''] > .section.active > [data-section-content], [data-section=''] > .section.active > .content, [data-section=''] > [data-section-region].active > [data-section-content], [data-section=''] > [data-section-region].active > .content, [data-section='auto'] > section.active > [data-section-content], [data-section='auto'] > section.active > .content, [data-section='auto'] > .section.active > [data-section-content], [data-section='auto'] > .section.active > .content, [data-section='auto'] > [data-section-region].active > [data-section-content], [data-section='auto'] > [data-section-region].active > .content, .section-container.auto > section.active > [data-section-content], .section-container.auto > section.active > .content, .section-container.auto > .section.active > [data-section-content], .section-container.auto > .section.active > .content, .section-container.auto > [data-section-region].active > [data-section-content], .section-container.auto > [data-section-region].active > .content {

display: block;

}

[data-section=''] > section:not(.active), [data-section=''] > .section:not(.active), [data-section=''] > [data-section-region]:not(.active), [data-section='auto'] > section:not(.active), [data-section='auto'] > .section:not(.active), [data-section='auto'] > [data-section-region]:not(.active), .section-container.auto > section:not(.active), .section-container.auto > .section:not(.active), .section-container.auto > [data-section-region]:not(.active) {

padding: 0 !important;

}

[data-section=''] > section > [data-section-title], [data-section=''] > section > .title, [data-section=''] > .section > [data-section-title], [data-section=''] > .section > .title, [data-section=''] > [data-section-region] > [data-section-title], [data-section=''] > [data-section-region] > .title, [data-section='auto'] > section > [data-section-title], [data-section='auto'] > section > .title, [data-section='auto'] > .section > [data-section-title], [data-section='auto'] > .section > .title, [data-section='auto'] > [data-section-region] > [data-section-title], [data-section='auto'] > [data-section-region] > .title, .section-container.auto > section > [data-section-title], .section-container.auto > section > .title, .section-container.auto > .section > [data-section-title], .section-container.auto > .section > .title, .section-container.auto > [data-section-region] > [data-section-title], .section-container.auto > [data-section-region] > .title {

width: auto;

position: absolute;

top: 0;

left: 0;

}

.section-container.auto {

border: none;

}

.section-container.auto > section > .title, .section-container.auto > .section > .title {

background-color: #efefef;

cursor: pointer;

border: solid 1px #cccccc;

}

.section-container.auto > section > .title a, .section-container.auto > .section > .title a {

padding: 0.9375em;

color: #333333;

font-size: 0.875em;

background: none;

}

.section-container.auto > section > .title:hover, .section-container.auto > .section > .title:hover {

background-color: #e2e2e2;

}

.section-container.auto > section > .content, .section-container.auto > .section > .content {

padding: 0.9375em;

background-color: white;

border: solid 1px #cccccc;

}

.section-container.auto > section > .content > \*:last-child, .section-container.auto > .section > .content > \*:last-child {

margin-bottom: 0;

}

.section-container.auto > section > .content > \*:first-child, .section-container.auto > .section > .content > \*:first-child {

padding-top: 0;

}

.section-container.auto > section > .content > \*:last-child:not(.flex-video), .section-container.auto > .section > .content > \*:last-child:not(.flex-video) {

padding-bottom: 0;

}

.section-container.auto > section.active > .title, .section-container.auto > .section.active > .title {

background: white;

}

.section-container.auto > section.active > .title a, .section-container.auto > .section.active > .title a {

color: #333333;

}

.section-container.auto > section:not(.active), .section-container.auto > .section:not(.active) {

padding: 0 !important;

}

.section-container.auto > section.active > .title, .section-container.auto > .section.active > .title {

border-bottom: 0;

}

[data-section='vertical-tabs'], .section-container.vertical-tabs {

width: 100%;

position: relative;

display: block;

margin-bottom: 1.25em;

}

[data-section='vertical-tabs']:not([data-section-resized]):not([data-section-small-style]), .section-container.vertical-tabs:not([data-section-resized]):not([data-section-small-style]) {

visibility: hidden;

}

[data-section='vertical-tabs'][data-section-small-style], .section-container.vertical-tabs[data-section-small-style] {

width: 100% !important;

}

[data-section='vertical-tabs'][data-section-small-style] > [data-section-region], [data-section='vertical-tabs'][data-section-small-style] > section, [data-section='vertical-tabs'][data-section-small-style] > .section, .section-container.vertical-tabs[data-section-small-style] > [data-section-region], .section-container.vertical-tabs[data-section-small-style] > section, .section-container.vertical-tabs[data-section-small-style] > .section {

padding: 0 !important;

margin: 0 !important;

}

[data-section='vertical-tabs'][data-section-small-style] > [data-section-region] > [data-section-title], [data-section='vertical-tabs'][data-section-small-style] > [data-section-region] > .title, [data-section='vertical-tabs'][data-section-small-style] > section > [data-section-title], [data-section='vertical-tabs'][data-section-small-style] > section > .title, [data-section='vertical-tabs'][data-section-small-style] > .section > [data-section-title], [data-section='vertical-tabs'][data-section-small-style] > .section > .title, .section-container.vertical-tabs[data-section-small-style] > [data-section-region] > [data-section-title], .section-container.vertical-tabs[data-section-small-style] > [data-section-region] > .title, .section-container.vertical-tabs[data-section-small-style] > section > [data-section-title], .section-container.vertical-tabs[data-section-small-style] > section > .title, .section-container.vertical-tabs[data-section-small-style] > .section > [data-section-title], .section-container.vertical-tabs[data-section-small-style] > .section > .title {

width: 100% !important;

}

[data-section='vertical-tabs'] > section > [data-section-title], [data-section='vertical-tabs'] > section > .title, [data-section='vertical-tabs'] > .section > [data-section-title], [data-section='vertical-tabs'] > .section > .title, [data-section='vertical-tabs'] > [data-section-region] > [data-section-title], [data-section='vertical-tabs'] > [data-section-region] > .title, .section-container.vertical-tabs > section > [data-section-title], .section-container.vertical-tabs > section > .title, .section-container.vertical-tabs > .section > [data-section-title], .section-container.vertical-tabs > .section > .title, .section-container.vertical-tabs > [data-section-region] > [data-section-title], .section-container.vertical-tabs > [data-section-region] > .title {

margin-bottom: 0;

}

[data-section='vertical-tabs'] > section > [data-section-title] a, [data-section='vertical-tabs'] > section > .title a, [data-section='vertical-tabs'] > .section > [data-section-title] a, [data-section='vertical-tabs'] > .section > .title a, [data-section='vertical-tabs'] > [data-section-region] > [data-section-title] a, [data-section='vertical-tabs'] > [data-section-region] > .title a, .section-container.vertical-tabs > section > [data-section-title] a, .section-container.vertical-tabs > section > .title a, .section-container.vertical-tabs > .section > [data-section-title] a, .section-container.vertical-tabs > .section > .title a, .section-container.vertical-tabs > [data-section-region] > [data-section-title] a, .section-container.vertical-tabs > [data-section-region] > .title a {

width: 100%;

display: inline-block;

white-space: nowrap;

}

[data-section='vertical-tabs'] > section > [data-section-content], [data-section='vertical-tabs'] > section > .content, [data-section='vertical-tabs'] > .section > [data-section-content], [data-section='vertical-tabs'] > .section > .content, [data-section='vertical-tabs'] > [data-section-region] > [data-section-content], [data-section='vertical-tabs'] > [data-section-region] > .content, .section-container.vertical-tabs > section > [data-section-content], .section-container.vertical-tabs > section > .content, .section-container.vertical-tabs > .section > [data-section-content], .section-container.vertical-tabs > .section > .content, .section-container.vertical-tabs > [data-section-region] > [data-section-content], .section-container.vertical-tabs > [data-section-region] > .content {

display: none;

}

[data-section='vertical-tabs'] > section.active > [data-section-content], [data-section='vertical-tabs'] > section.active > .content, [data-section='vertical-tabs'] > .section.active > [data-section-content], [data-section='vertical-tabs'] > .section.active > .content, [data-section='vertical-tabs'] > [data-section-region].active > [data-section-content], [data-section='vertical-tabs'] > [data-section-region].active > .content, .section-container.vertical-tabs > section.active > [data-section-content], .section-container.vertical-tabs > section.active > .content, .section-container.vertical-tabs > .section.active > [data-section-content], .section-container.vertical-tabs > .section.active > .content, .section-container.vertical-tabs > [data-section-region].active > [data-section-content], .section-container.vertical-tabs > [data-section-region].active > .content {

display: block;

}

[data-section='vertical-tabs'] > section:not(.active), [data-section='vertical-tabs'] > .section:not(.active), [data-section='vertical-tabs'] > [data-section-region]:not(.active), .section-container.vertical-tabs > section:not(.active), .section-container.vertical-tabs > .section:not(.active), .section-container.vertical-tabs > [data-section-region]:not(.active) {

padding: 0 !important;

}

[data-section='vertical-tabs'] > section > [data-section-title], [data-section='vertical-tabs'] > section > .title, [data-section='vertical-tabs'] > .section > [data-section-title], [data-section='vertical-tabs'] > .section > .title, [data-section='vertical-tabs'] > [data-section-region] > [data-section-title], [data-section='vertical-tabs'] > [data-section-region] > .title, .section-container.vertical-tabs > section > [data-section-title], .section-container.vertical-tabs > section > .title, .section-container.vertical-tabs > .section > [data-section-title], .section-container.vertical-tabs > .section > .title, .section-container.vertical-tabs > [data-section-region] > [data-section-title], .section-container.vertical-tabs > [data-section-region] > .title {

position: absolute;

top: 0;

left: 0;

width: 12.5em;

}

[data-section='vertical-tabs'] > section.active, [data-section='vertical-tabs'] > .section.active, [data-section='vertical-tabs'] > [data-section-region].active, .section-container.vertical-tabs > section.active, .section-container.vertical-tabs > .section.active, .section-container.vertical-tabs > [data-section-region].active {

padding-left: 12.5em;

}

[data-section='vertical-tabs'] > section.active > [data-section-title], [data-section='vertical-tabs'] > section.active > .title, [data-section='vertical-tabs'] > .section.active > [data-section-title], [data-section='vertical-tabs'] > .section.active > .title, [data-section='vertical-tabs'] > [data-section-region].active > [data-section-title], [data-section='vertical-tabs'] > [data-section-region].active > .title, .section-container.vertical-tabs > section.active > [data-section-title], .section-container.vertical-tabs > section.active > .title, .section-container.vertical-tabs > .section.active > [data-section-title], .section-container.vertical-tabs > .section.active > .title, .section-container.vertical-tabs > [data-section-region].active > [data-section-title], .section-container.vertical-tabs > [data-section-region].active > .title {

width: 12.5em;

}

.section-container.vertical-tabs {

border: none;

}

.section-container.vertical-tabs > section > .title, .section-container.vertical-tabs > .section > .title {

background-color: #efefef;

cursor: pointer;

border: solid 1px #cccccc;

}

.section-container.vertical-tabs > section > .title a, .section-container.vertical-tabs > .section > .title a {

padding: 0.9375em;

color: #333333;

font-size: 0.875em;

background: none;

}

.section-container.vertical-tabs > section > .title:hover, .section-container.vertical-tabs > .section > .title:hover {

background-color: #e2e2e2;

}

.section-container.vertical-tabs > section > .content, .section-container.vertical-tabs > .section > .content {

padding: 0.9375em;

background-color: white;

border: solid 1px #cccccc;

}

.section-container.vertical-tabs > section > .content > \*:last-child, .section-container.vertical-tabs > .section > .content > \*:last-child {

margin-bottom: 0;

}

.section-container.vertical-tabs > section > .content > \*:first-child, .section-container.vertical-tabs > .section > .content > \*:first-child {

padding-top: 0;

}

.section-container.vertical-tabs > section > .content > \*:last-child:not(.flex-video), .section-container.vertical-tabs > .section > .content > \*:last-child:not(.flex-video) {

padding-bottom: 0;

}

.section-container.vertical-tabs > section.active > .title, .section-container.vertical-tabs > .section.active > .title {

background: #d5d5d5;

}

.section-container.vertical-tabs > section.active > .title a, .section-container.vertical-tabs > .section.active > .title a {

color: #333333;

}

.section-container.vertical-tabs > section:not(.active), .section-container.vertical-tabs > .section:not(.active) {

padding: 0 !important;

}

.section-container.vertical-tabs > section.active, .section-container.vertical-tabs > .section.active {

padding-left: 12.4375em;

}

.section-container.vertical-tabs > section.active > .title, .section-container.vertical-tabs > .section.active > .title {

background-color: #d5d5d5;

}

[data-section='vertical-nav'], .section-container.vertical-nav {

width: 100%;

position: relative;

display: block;

margin-bottom: 1.25em;

}

[data-section='vertical-nav']:not([data-section-resized]):not([data-section-small-style]), .section-container.vertical-nav:not([data-section-resized]):not([data-section-small-style]) {

visibility: hidden;

}

[data-section='vertical-nav'][data-section-small-style], .section-container.vertical-nav[data-section-small-style] {

width: 100% !important;

}

[data-section='vertical-nav'][data-section-small-style] > [data-section-region], [data-section='vertical-nav'][data-section-small-style] > section, [data-section='vertical-nav'][data-section-small-style] > .section, .section-container.vertical-nav[data-section-small-style] > [data-section-region], .section-container.vertical-nav[data-section-small-style] > section, .section-container.vertical-nav[data-section-small-style] > .section {

padding: 0 !important;

margin: 0 !important;

}

[data-section='vertical-nav'][data-section-small-style] > [data-section-region] > [data-section-title], [data-section='vertical-nav'][data-section-small-style] > [data-section-region] > .title, [data-section='vertical-nav'][data-section-small-style] > section > [data-section-title], [data-section='vertical-nav'][data-section-small-style] > section > .title, [data-section='vertical-nav'][data-section-small-style] > .section > [data-section-title], [data-section='vertical-nav'][data-section-small-style] > .section > .title, .section-container.vertical-nav[data-section-small-style] > [data-section-region] > [data-section-title], .section-container.vertical-nav[data-section-small-style] > [data-section-region] > .title, .section-container.vertical-nav[data-section-small-style] > section > [data-section-title], .section-container.vertical-nav[data-section-small-style] > section > .title, .section-container.vertical-nav[data-section-small-style] > .section > [data-section-title], .section-container.vertical-nav[data-section-small-style] > .section > .title {

width: 100% !important;

}

[data-section='vertical-nav'] > section, [data-section='vertical-nav'] > .section, [data-section='vertical-nav'] > [data-section-region], .section-container.vertical-nav > section, .section-container.vertical-nav > .section, .section-container.vertical-nav > [data-section-region] {

position: relative;

display: inline-block;

}

[data-section='vertical-nav'] > section > [data-section-title], [data-section='vertical-nav'] > section > .title, [data-section='vertical-nav'] > .section > [data-section-title], [data-section='vertical-nav'] > .section > .title, [data-section='vertical-nav'] > [data-section-region] > [data-section-title], [data-section='vertical-nav'] > [data-section-region] > .title, .section-container.vertical-nav > section > [data-section-title], .section-container.vertical-nav > section > .title, .section-container.vertical-nav > .section > [data-section-title], .section-container.vertical-nav > .section > .title, .section-container.vertical-nav > [data-section-region] > [data-section-title], .section-container.vertical-nav > [data-section-region] > .title {

margin-bottom: 0;

}

[data-section='vertical-nav'] > section > [data-section-title] a, [data-section='vertical-nav'] > section > .title a, [data-section='vertical-nav'] > .section > [data-section-title] a, [data-section='vertical-nav'] > .section > .title a, [data-section='vertical-nav'] > [data-section-region] > [data-section-title] a, [data-section='vertical-nav'] > [data-section-region] > .title a, .section-container.vertical-nav > section > [data-section-title] a, .section-container.vertical-nav > section > .title a, .section-container.vertical-nav > .section > [data-section-title] a, .section-container.vertical-nav > .section > .title a, .section-container.vertical-nav > [data-section-region] > [data-section-title] a, .section-container.vertical-nav > [data-section-region] > .title a {

width: 100%;

display: inline-block;

white-space: nowrap;

}

[data-section='vertical-nav'] > section > [data-section-content], [data-section='vertical-nav'] > section > .content, [data-section='vertical-nav'] > .section > [data-section-content], [data-section='vertical-nav'] > .section > .content, [data-section='vertical-nav'] > [data-section-region] > [data-section-content], [data-section='vertical-nav'] > [data-section-region] > .content, .section-container.vertical-nav > section > [data-section-content], .section-container.vertical-nav > section > .content, .section-container.vertical-nav > .section > [data-section-content], .section-container.vertical-nav > .section > .content, .section-container.vertical-nav > [data-section-region] > [data-section-content], .section-container.vertical-nav > [data-section-region] > .content {

display: none;

}

[data-section='vertical-nav'] > section.active > [data-section-content], [data-section='vertical-nav'] > section.active > .content, [data-section='vertical-nav'] > .section.active > [data-section-content], [data-section='vertical-nav'] > .section.active > .content, [data-section='vertical-nav'] > [data-section-region].active > [data-section-content], [data-section='vertical-nav'] > [data-section-region].active > .content, .section-container.vertical-nav > section.active > [data-section-content], .section-container.vertical-nav > section.active > .content, .section-container.vertical-nav > .section.active > [data-section-content], .section-container.vertical-nav > .section.active > .content, .section-container.vertical-nav > [data-section-region].active > [data-section-content], .section-container.vertical-nav > [data-section-region].active > .content {

display: block;

}

[data-section='vertical-nav'] > section:not(.active), [data-section='vertical-nav'] > .section:not(.active), [data-section='vertical-nav'] > [data-section-region]:not(.active), .section-container.vertical-nav > section:not(.active), .section-container.vertical-nav > .section:not(.active), .section-container.vertical-nav > [data-section-region]:not(.active) {

padding: 0 !important;

}

[data-section='vertical-nav'] > section > [data-section-title], [data-section='vertical-nav'] > section > .title, [data-section='vertical-nav'] > .section > [data-section-title], [data-section='vertical-nav'] > .section > .title, [data-section='vertical-nav'] > [data-section-region] > [data-section-title], [data-section='vertical-nav'] > [data-section-region] > .title, .section-container.vertical-nav > section > [data-section-title], .section-container.vertical-nav > section > .title, .section-container.vertical-nav > .section > [data-section-title], .section-container.vertical-nav > .section > .title, .section-container.vertical-nav > [data-section-region] > [data-section-title], .section-container.vertical-nav > [data-section-region] > .title {

position: static;

width: auto;

}

[data-section='vertical-nav'] > section > [data-section-title] a, [data-section='vertical-nav'] > section > .title a, [data-section='vertical-nav'] > .section > [data-section-title] a, [data-section='vertical-nav'] > .section > .title a, [data-section='vertical-nav'] > [data-section-region] > [data-section-title] a, [data-section='vertical-nav'] > [data-section-region] > .title a, .section-container.vertical-nav > section > [data-section-title] a, .section-container.vertical-nav > section > .title a, .section-container.vertical-nav > .section > [data-section-title] a, .section-container.vertical-nav > .section > .title a, .section-container.vertical-nav > [data-section-region] > [data-section-title] a, .section-container.vertical-nav > [data-section-region] > .title a {

display: block;

}

[data-section='vertical-nav'] > section > [data-section-content], [data-section='vertical-nav'] > section > .content, [data-section='vertical-nav'] > .section > [data-section-content], [data-section='vertical-nav'] > .section > .content, [data-section='vertical-nav'] > [data-section-region] > [data-section-content], [data-section='vertical-nav'] > [data-section-region] > .content, .section-container.vertical-nav > section > [data-section-content], .section-container.vertical-nav > section > .content, .section-container.vertical-nav > .section > [data-section-content], .section-container.vertical-nav > .section > .content, .section-container.vertical-nav > [data-section-region] > [data-section-content], .section-container.vertical-nav > [data-section-region] > .content {

position: absolute;

top: 0;

left: 0;

z-index: 999;

min-width: 12.5em;

}

.section-container.vertical-nav {

border: none;

}

.section-container.vertical-nav > section > .title, .section-container.vertical-nav > .section > .title {

background-color: #efefef;

cursor: pointer;

border: solid 1px #cccccc;

}

.section-container.vertical-nav > section > .title a, .section-container.vertical-nav > .section > .title a {

padding: 0.9375em;

color: #333333;

font-size: 0.875em;

background: none;

}

.section-container.vertical-nav > section > .title:hover, .section-container.vertical-nav > .section > .title:hover {

background-color: #e2e2e2;

}

.section-container.vertical-nav > section > .content, .section-container.vertical-nav > .section > .content {

padding: 0.9375em;

background-color: white;

border: solid 1px #cccccc;

}

.section-container.vertical-nav > section > .content > \*:last-child, .section-container.vertical-nav > .section > .content > \*:last-child {

margin-bottom: 0;

}

.section-container.vertical-nav > section > .content > \*:first-child, .section-container.vertical-nav > .section > .content > \*:first-child {

padding-top: 0;

}

.section-container.vertical-nav > section > .content > \*:last-child:not(.flex-video), .section-container.vertical-nav > .section > .content > \*:last-child:not(.flex-video) {

padding-bottom: 0;

}

.section-container.vertical-nav > section.active > .title, .section-container.vertical-nav > .section.active > .title {

background: #d5d5d5;

}

.section-container.vertical-nav > section.active > .title a, .section-container.vertical-nav > .section.active > .title a {

color: #333333;

}

.section-container.vertical-nav > section:not(.active), .section-container.vertical-nav > .section:not(.active) {

padding: 0 !important;

}

[data-section='horizontal-nav'], .section-container.horizontal-nav {

width: 100%;

position: relative;

display: block;

margin-bottom: 1.25em;

}

[data-section='horizontal-nav']:not([data-section-resized]):not([data-section-small-style]), .section-container.horizontal-nav:not([data-section-resized]):not([data-section-small-style]) {

visibility: hidden;

}

[data-section='horizontal-nav'][data-section-small-style], .section-container.horizontal-nav[data-section-small-style] {

width: 100% !important;

}

[data-section='horizontal-nav'][data-section-small-style] > [data-section-region], [data-section='horizontal-nav'][data-section-small-style] > section, [data-section='horizontal-nav'][data-section-small-style] > .section, .section-container.horizontal-nav[data-section-small-style] > [data-section-region], .section-container.horizontal-nav[data-section-small-style] > section, .section-container.horizontal-nav[data-section-small-style] > .section {

padding: 0 !important;

margin: 0 !important;

}

[data-section='horizontal-nav'][data-section-small-style] > [data-section-region] > [data-section-title], [data-section='horizontal-nav'][data-section-small-style] > [data-section-region] > .title, [data-section='horizontal-nav'][data-section-small-style] > section > [data-section-title], [data-section='horizontal-nav'][data-section-small-style] > section > .title, [data-section='horizontal-nav'][data-section-small-style] > .section > [data-section-title], [data-section='horizontal-nav'][data-section-small-style] > .section > .title, .section-container.horizontal-nav[data-section-small-style] > [data-section-region] > [data-section-title], .section-container.horizontal-nav[data-section-small-style] > [data-section-region] > .title, .section-container.horizontal-nav[data-section-small-style] > section > [data-section-title], .section-container.horizontal-nav[data-section-small-style] > section > .title, .section-container.horizontal-nav[data-section-small-style] > .section > [data-section-title], .section-container.horizontal-nav[data-section-small-style] > .section > .title {

width: 100% !important;

}

[data-section='horizontal-nav'] > section, [data-section='horizontal-nav'] > .section, [data-section='horizontal-nav'] > [data-section-region], .section-container.horizontal-nav > section, .section-container.horizontal-nav > .section, .section-container.horizontal-nav > [data-section-region] {

position: relative;

float: left;

}

[data-section='horizontal-nav'] > section > [data-section-title], [data-section='horizontal-nav'] > section > .title, [data-section='horizontal-nav'] > .section > [data-section-title], [data-section='horizontal-nav'] > .section > .title, [data-section='horizontal-nav'] > [data-section-region] > [data-section-title], [data-section='horizontal-nav'] > [data-section-region] > .title, .section-container.horizontal-nav > section > [data-section-title], .section-container.horizontal-nav > section > .title, .section-container.horizontal-nav > .section > [data-section-title], .section-container.horizontal-nav > .section > .title, .section-container.horizontal-nav > [data-section-region] > [data-section-title], .section-container.horizontal-nav > [data-section-region] > .title {

margin-bottom: 0;

}

[data-section='horizontal-nav'] > section > [data-section-title] a, [data-section='horizontal-nav'] > section > .title a, [data-section='horizontal-nav'] > .section > [data-section-title] a, [data-section='horizontal-nav'] > .section > .title a, [data-section='horizontal-nav'] > [data-section-region] > [data-section-title] a, [data-section='horizontal-nav'] > [data-section-region] > .title a, .section-container.horizontal-nav > section > [data-section-title] a, .section-container.horizontal-nav > section > .title a, .section-container.horizontal-nav > .section > [data-section-title] a, .section-container.horizontal-nav > .section > .title a, .section-container.horizontal-nav > [data-section-region] > [data-section-title] a, .section-container.horizontal-nav > [data-section-region] > .title a {

width: 100%;

display: inline-block;

white-space: nowrap;

}

[data-section='horizontal-nav'] > section > [data-section-content], [data-section='horizontal-nav'] > section > .content, [data-section='horizontal-nav'] > .section > [data-section-content], [data-section='horizontal-nav'] > .section > .content, [data-section='horizontal-nav'] > [data-section-region] > [data-section-content], [data-section='horizontal-nav'] > [data-section-region] > .content, .section-container.horizontal-nav > section > [data-section-content], .section-container.horizontal-nav > section > .content, .section-container.horizontal-nav > .section > [data-section-content], .section-container.horizontal-nav > .section > .content, .section-container.horizontal-nav > [data-section-region] > [data-section-content], .section-container.horizontal-nav > [data-section-region] > .content {

display: none;

}

[data-section='horizontal-nav'] > section.active > [data-section-content], [data-section='horizontal-nav'] > section.active > .content, [data-section='horizontal-nav'] > .section.active > [data-section-content], [data-section='horizontal-nav'] > .section.active > .content, [data-section='horizontal-nav'] > [data-section-region].active > [data-section-content], [data-section='horizontal-nav'] > [data-section-region].active > .content, .section-container.horizontal-nav > section.active > [data-section-content], .section-container.horizontal-nav > section.active > .content, .section-container.horizontal-nav > .section.active > [data-section-content], .section-container.horizontal-nav > .section.active > .content, .section-container.horizontal-nav > [data-section-region].active > [data-section-content], .section-container.horizontal-nav > [data-section-region].active > .content {

display: block;

}

[data-section='horizontal-nav'] > section:not(.active), [data-section='horizontal-nav'] > .section:not(.active), [data-section='horizontal-nav'] > [data-section-region]:not(.active), .section-container.horizontal-nav > section:not(.active), .section-container.horizontal-nav > .section:not(.active), .section-container.horizontal-nav > [data-section-region]:not(.active) {

padding: 0 !important;

}

[data-section='horizontal-nav'] > section > [data-section-title], [data-section='horizontal-nav'] > section > .title, [data-section='horizontal-nav'] > .section > [data-section-title], [data-section='horizontal-nav'] > .section > .title, [data-section='horizontal-nav'] > [data-section-region] > [data-section-title], [data-section='horizontal-nav'] > [data-section-region] > .title, .section-container.horizontal-nav > section > [data-section-title], .section-container.horizontal-nav > section > .title, .section-container.horizontal-nav > .section > [data-section-title], .section-container.horizontal-nav > .section > .title, .section-container.horizontal-nav > [data-section-region] > [data-section-title], .section-container.horizontal-nav > [data-section-region] > .title {

position: static;

width: auto;

}

[data-section='horizontal-nav'] > section > [data-section-title] a, [data-section='horizontal-nav'] > section > .title a, [data-section='horizontal-nav'] > .section > [data-section-title] a, [data-section='horizontal-nav'] > .section > .title a, [data-section='horizontal-nav'] > [data-section-region] > [data-section-title] a, [data-section='horizontal-nav'] > [data-section-region] > .title a, .section-container.horizontal-nav > section > [data-section-title] a, .section-container.horizontal-nav > section > .title a, .section-container.horizontal-nav > .section > [data-section-title] a, .section-container.horizontal-nav > .section > .title a, .section-container.horizontal-nav > [data-section-region] > [data-section-title] a, .section-container.horizontal-nav > [data-section-region] > .title a {

display: block;

}

[data-section='horizontal-nav'] > section > [data-section-content], [data-section='horizontal-nav'] > section > .content, [data-section='horizontal-nav'] > .section > [data-section-content], [data-section='horizontal-nav'] > .section > .content, [data-section='horizontal-nav'] > [data-section-region] > [data-section-content], [data-section='horizontal-nav'] > [data-section-region] > .content, .section-container.horizontal-nav > section > [data-section-content], .section-container.horizontal-nav > section > .content, .section-container.horizontal-nav > .section > [data-section-content], .section-container.horizontal-nav > .section > .content, .section-container.horizontal-nav > [data-section-region] > [data-section-content], .section-container.horizontal-nav > [data-section-region] > .content {

width: auto;

position: absolute;

top: 0;

left: 0;

z-index: 999;

min-width: 12.5em;

}

.section-container.horizontal-nav {

background: #efefef;

border: 1px solid #cccccc;

}

.section-container.horizontal-nav > section > .title, .section-container.horizontal-nav > .section > .title {

background-color: #efefef;

cursor: pointer;

border: solid 1px #cccccc;

}

.section-container.horizontal-nav > section > .title a, .section-container.horizontal-nav > .section > .title a {

padding: 0.9375em;

color: #333333;

font-size: 0.875em;

background: none;

}

.section-container.horizontal-nav > section > .title:hover, .section-container.horizontal-nav > .section > .title:hover {

background-color: #e2e2e2;

}

.section-container.horizontal-nav > section > .content, .section-container.horizontal-nav > .section > .content {

padding: 0.9375em;

background-color: white;

border: solid 1px #cccccc;

}

.section-container.horizontal-nav > section > .content > \*:last-child, .section-container.horizontal-nav > .section > .content > \*:last-child {

margin-bottom: 0;

}

.section-container.horizontal-nav > section > .content > \*:first-child, .section-container.horizontal-nav > .section > .content > \*:first-child {

padding-top: 0;

}

.section-container.horizontal-nav > section > .content > \*:last-child:not(.flex-video), .section-container.horizontal-nav > .section > .content > \*:last-child:not(.flex-video) {

padding-bottom: 0;

}

.section-container.horizontal-nav > section.active > .title, .section-container.horizontal-nav > .section.active > .title {

background: #d5d5d5;

}

.section-container.horizontal-nav > section.active > .title a, .section-container.horizontal-nav > .section.active > .title a {

color: #333333;

}

.section-container.horizontal-nav > section:not(.active), .section-container.horizontal-nav > .section:not(.active) {

padding: 0 !important;

}

}

.no-js [data-section], .no-js .section-container {

width: 100%;

position: relative;

display: block;

margin-bottom: 1.25em;

}

.no-js [data-section][data-section-small-style], .no-js .section-container[data-section-small-style] {

width: 100% !important;

}

.no-js [data-section][data-section-small-style] > [data-section-region], .no-js [data-section][data-section-small-style] > section, .no-js [data-section][data-section-small-style] > .section, .no-js .section-container[data-section-small-style] > [data-section-region], .no-js .section-container[data-section-small-style] > section, .no-js .section-container[data-section-small-style] > .section {

padding: 0 !important;

margin: 0 !important;

}

.no-js [data-section][data-section-small-style] > [data-section-region] > [data-section-title], .no-js [data-section][data-section-small-style] > [data-section-region] > .title, .no-js [data-section][data-section-small-style] > section > [data-section-title], .no-js [data-section][data-section-small-style] > section > .title, .no-js [data-section][data-section-small-style] > .section > [data-section-title], .no-js [data-section][data-section-small-style] > .section > .title, .no-js .section-container[data-section-small-style] > [data-section-region] > [data-section-title], .no-js .section-container[data-section-small-style] > [data-section-region] > .title, .no-js .section-container[data-section-small-style] > section > [data-section-title], .no-js .section-container[data-section-small-style] > section > .title, .no-js .section-container[data-section-small-style] > .section > [data-section-title], .no-js .section-container[data-section-small-style] > .section > .title {

width: 100% !important;

}

.no-js [data-section] > section, .no-js [data-section] > .section, .no-js [data-section] > [data-section-region], .no-js .section-container > section, .no-js .section-container > .section, .no-js .section-container > [data-section-region] {

margin: 0;

}

.no-js [data-section] > section > [data-section-title], .no-js [data-section] > section > .title, .no-js [data-section] > .section > [data-section-title], .no-js [data-section] > .section > .title, .no-js [data-section] > [data-section-region] > [data-section-title], .no-js [data-section] > [data-section-region] > .title, .no-js .section-container > section > [data-section-title], .no-js .section-container > section > .title, .no-js .section-container > .section > [data-section-title], .no-js .section-container > .section > .title, .no-js .section-container > [data-section-region] > [data-section-title], .no-js .section-container > [data-section-region] > .title {

margin-bottom: 0;

}

.no-js [data-section] > section > [data-section-title] a, .no-js [data-section] > section > .title a, .no-js [data-section] > .section > [data-section-title] a, .no-js [data-section] > .section > .title a, .no-js [data-section] > [data-section-region] > [data-section-title] a, .no-js [data-section] > [data-section-region] > .title a, .no-js .section-container > section > [data-section-title] a, .no-js .section-container > section > .title a, .no-js .section-container > .section > [data-section-title] a, .no-js .section-container > .section > .title a, .no-js .section-container > [data-section-region] > [data-section-title] a, .no-js .section-container > [data-section-region] > .title a {

width: 100%;

display: inline-block;

white-space: nowrap;

}

.no-js [data-section] > section > [data-section-content], .no-js [data-section] > section > .content, .no-js [data-section] > .section > [data-section-content], .no-js [data-section] > .section > .content, .no-js [data-section] > [data-section-region] > [data-section-content], .no-js [data-section] > [data-section-region] > .content, .no-js .section-container > section > [data-section-content], .no-js .section-container > section > .content, .no-js .section-container > .section > [data-section-content], .no-js .section-container > .section > .content, .no-js .section-container > [data-section-region] > [data-section-content], .no-js .section-container > [data-section-region] > .content {

display: none;

}

.no-js [data-section] > section.active > [data-section-content], .no-js [data-section] > section.active > .content, .no-js [data-section] > .section.active > [data-section-content], .no-js [data-section] > .section.active > .content, .no-js [data-section] > [data-section-region].active > [data-section-content], .no-js [data-section] > [data-section-region].active > .content, .no-js .section-container > section.active > [data-section-content], .no-js .section-container > section.active > .content, .no-js .section-container > .section.active > [data-section-content], .no-js .section-container > .section.active > .content, .no-js .section-container > [data-section-region].active > [data-section-content], .no-js .section-container > [data-section-region].active > .content {

display: block;

}

.no-js [data-section] > section:not(.active), .no-js [data-section] > .section:not(.active), .no-js [data-section] > [data-section-region]:not(.active), .no-js .section-container > section:not(.active), .no-js .section-container > .section:not(.active), .no-js .section-container > [data-section-region]:not(.active) {

padding: 0 !important;

}

.no-js [data-section] > section > [data-section-title], .no-js [data-section] > section > .title, .no-js [data-section] > .section > [data-section-title], .no-js [data-section] > .section > .title, .no-js [data-section] > [data-section-region] > [data-section-title], .no-js [data-section] > [data-section-region] > .title, .no-js .section-container > section > [data-section-title], .no-js .section-container > section > .title, .no-js .section-container > .section > [data-section-title], .no-js .section-container > .section > .title, .no-js .section-container > [data-section-region] > [data-section-title], .no-js .section-container > [data-section-region] > .title {

width: 100%;

}

.no-js .section-container {

border-top: 1px solid #cccccc;

}

.no-js .section-container > section > .title, .no-js .section-container > .section > .title {

background-color: #efefef;

cursor: pointer;

border: solid 1px #cccccc;

}

.no-js .section-container > section > .title a, .no-js .section-container > .section > .title a {

padding: 0.9375em;

color: #333333;

font-size: 0.875em;

background: none;

}

.no-js .section-container > section > .title:hover, .no-js .section-container > .section > .title:hover {

background-color: #e2e2e2;

}

.no-js .section-container > section > .content, .no-js .section-container > .section > .content {

padding: 0.9375em;

background-color: white;

border: solid 1px #cccccc;

}

.no-js .section-container > section > .content > \*:last-child, .no-js .section-container > .section > .content > \*:last-child {

margin-bottom: 0;

}

.no-js .section-container > section > .content > \*:first-child, .no-js .section-container > .section > .content > \*:first-child {

padding-top: 0;

}

.no-js .section-container > section > .content > \*:last-child:not(.flex-video), .no-js .section-container > .section > .content > \*:last-child:not(.flex-video) {

padding-bottom: 0;

}

.no-js .section-container > section.active > .title, .no-js .section-container > .section.active > .title {

background: #d5d5d5;

}

.no-js .section-container > section.active > .title a, .no-js .section-container > .section.active > .title a {

color: #333333;

}

.no-js .section-container > section:not(.active), .no-js .section-container > .section:not(.active) {

padding: 0 !important;

}

.no-js .section-container > section > .title, .no-js .section-container > .section > .title {

border-top: none;

}

/\* Wrapped around .top-bar to contain to grid width \*/

.contain-to-grid {

width: 100%;

background: #111111;

}

.contain-to-grid .top-bar {

margin-bottom: 1.875em;

}

.fixed {

width: 100%;

left: 0;

position: fixed;

top: 0;

z-index: 99;

}

.top-bar {

overflow: hidden;

height: 45px;

line-height: 45px;

position: relative;

background: #111111;

margin-bottom: 1.875em;

}

.top-bar ul {

margin-bottom: 0;

list-style: none;

}

.top-bar .row {

max-width: none;

}

.top-bar form,

.top-bar input {

margin-bottom: 0;

}

.top-bar input {

height: 2.45em;

}

.top-bar .button {

padding-top: .5em;

padding-bottom: .5em;

margin-bottom: 0;

}

.top-bar .title-area {

position: relative;

margin: 0;

}

.top-bar .name {

height: 45px;

margin: 0;

font-size: 16;

}

.top-bar .name h1 {

line-height: 45px;

font-size: 1.0625em;

margin: 0;

}

.top-bar .name h1 a {

font-weight: bold;

color: white;

width: 50%;

display: block;

padding: 0 15px;

}

.top-bar .toggle-topbar {

position: absolute;

right: 0;

top: 0;

}

.top-bar .toggle-topbar a {

color: white;

text-transform: uppercase;

font-size: 0.8125em;

font-weight: bold;

position: relative;

display: block;

padding: 0 15px;

height: 45px;

line-height: 45px;

}

.top-bar .toggle-topbar.menu-icon {

right: 15px;

top: 50%;

margin-top: -16px;

padding-left: 40px;

}

.top-bar .toggle-topbar.menu-icon a {

text-indent: -48px;

width: 34px;

height: 34px;

line-height: 33px;

padding: 0;

color: white;

}

.top-bar .toggle-topbar.menu-icon a span {

position: absolute;

right: 0;

display: block;

width: 16px;

height: 0;

-webkit-box-shadow: 0 10px 0 1px white, 0 16px 0 1px white, 0 22px 0 1px white;

box-shadow: 0 10px 0 1px white, 0 16px 0 1px white, 0 22px 0 1px white;

}

.top-bar.expanded {

height: auto;

background: transparent;

}

.top-bar.expanded .title-area {

background: #111111;

}

.top-bar.expanded .toggle-topbar a {

color: #888888;

}

.top-bar.expanded .toggle-topbar a span {

-webkit-box-shadow: 0 10px 0 1px #888888, 0 16px 0 1px #888888, 0 22px 0 1px #888888;

box-shadow: 0 10px 0 1px #888888, 0 16px 0 1px #888888, 0 22px 0 1px #888888;

}

.top-bar-section {

left: 0;

position: relative;

width: auto;

-webkit-transition: left 300ms ease-out;

-moz-transition: left 300ms ease-out;

transition: left 300ms ease-out;

}

.top-bar-section ul {

width: 100%;

height: auto;

display: block;

background: #222222;

font-size: 16;

margin: 0;

}

.top-bar-section .divider,

.top-bar-section [role="separator"] {

border-bottom: solid 1px #2b2b2b;

border-top: solid 1px black;

clear: both;

height: 1px;

width: 100%;

}

.top-bar-section ul li > a {

display: block;

width: 100%;

color: white;

padding: 12px 0 12px 0;

padding-left: 15px;

font-size: 0.8125em;

font-weight: bold;

background: #222222;

}

.top-bar-section ul li > a.button {

background: #2ba6cb;

font-size: 0.8125em;

}

.top-bar-section ul li > a.button.hover {

background: #2284a1;

}

.top-bar-section ul li > a.button.secondary {

background: #e9e9e9;

}

.top-bar-section ul li > a.button.secondary.hover {

background: #d0d0d0;

}

.top-bar-section ul li > a.button.success {

background: #5da423;

}

.top-bar-section ul li > a.button.success.hover {

background: #457a1a;

}

.top-bar-section ul li > a.button.alert {

background: #c60f13;

}

.top-bar-section ul li > a.button.alert.hover {

background: #970b0e;

}

.top-bar-section ul li.hover > a {

background: #090909;

color: white;

}

.top-bar-section ul li.active > a {

background: #090909;

color: white;

}

.top-bar-section .has-form {

padding: 15px;

}

.top-bar-section .has-dropdown {

position: relative;

}

.top-bar-section .has-dropdown > a:after {

content: "";

display: block;

width: 0;

height: 0;

border: inset 5px;

border-color: transparent transparent transparent rgba(255, 255, 255, 0.5);

border-left-style: solid;

margin-right: 15px;

margin-top: -4.5px;

position: absolute;

top: 50%;

right: 0;

}

.top-bar-section .has-dropdown.moved {

position: static;

}

.top-bar-section .has-dropdown.moved > .dropdown {

visibility: visible;

}

.top-bar-section .dropdown {

position: absolute;

left: 100%;

top: 0;

visibility: hidden;

z-index: 99;

}

.top-bar-section .dropdown li {

width: 100%;

height: auto;

}

.top-bar-section .dropdown li a {

font-weight: normal;

padding: 8px 15px;

}

.top-bar-section .dropdown li a.parent-link {

font-weight: bold;

}

.top-bar-section .dropdown li.title h5 {

margin-bottom: 0;

}

.top-bar-section .dropdown li.title h5 a {

color: white;

line-height: 22.5px;

display: block;

}

.top-bar-section .dropdown label {

padding: 8px 15px 2px;

margin-bottom: 0;

text-transform: uppercase;

color: #555555;

font-weight: bold;

font-size: 0.625em;

}

.top-bar-js-breakpoint {

width: 58.75em !important;

visibility: hidden;

}

.js-generated {

display: block;

}

@media only screen and (min-width: 58.75em) {

.top-bar {

background: #111111;

\*zoom: 1;

overflow: visible;

}

.top-bar:before, .top-bar:after {

content: " ";

display: table;

}

.top-bar:after {

clear: both;

}

.top-bar .toggle-topbar {

display: none;

}

.top-bar .title-area {

float: left;

}

.top-bar .name h1 a {

width: auto;

}

.top-bar input,

.top-bar .button {

line-height: 2em;

font-size: 0.875em;

height: 2em;

padding: 0 10px;

position: relative;

top: 8px;

}

.top-bar.expanded {

background: #111111;

}

.contain-to-grid .top-bar {

max-width: 62.5em;

margin: 0 auto;

margin-bottom: 1.875em;

}

.top-bar-section {

-webkit-transition: none 0 0;

-moz-transition: none 0 0;

transition: none 0 0;

left: 0 !important;

}

.top-bar-section ul {

width: auto;

height: auto !important;

display: inline;

}

.top-bar-section ul li {

float: left;

}

.top-bar-section ul li .js-generated {

display: none;

}

.top-bar-section li a:not(.button) {

padding: 0 15px;

line-height: 45px;

background: #111111;

}

.top-bar-section li a:not(.button).hover {

background: black;

}

.top-bar-section .has-dropdown > a {

padding-right: 35px !important;

}

.top-bar-section .has-dropdown > a:after {

content: "";

display: block;

width: 0;

height: 0;

border: inset 5px;

border-color: rgba(255, 255, 255, 0.5) transparent transparent transparent;

border-top-style: solid;

margin-top: -2.5px;

top: 22.5px;

}

.top-bar-section .has-dropdown.moved {

position: relative;

}

.top-bar-section .has-dropdown.moved > .dropdown {

visibility: hidden;

}

.top-bar-section .has-dropdown.hover > .dropdown, .top-bar-section .has-dropdown.not-click:hover > .dropdown {

visibility: visible;

}

.top-bar-section .has-dropdown .dropdown li.has-dropdown > a:after {

border: none;

content: "\00bb";

margin-top: -16px;

right: 5px;

}

.top-bar-section .dropdown {

left: 0;

top: auto;

background: transparent;

min-width: 100%;

}

.top-bar-section .dropdown li a {

color: white;

line-height: 1;

white-space: nowrap;

padding: 7px 15px;

background: #1e1e1e;

}

.top-bar-section .dropdown li label {

white-space: nowrap;

background: #1e1e1e;

}

.top-bar-section .dropdown li .dropdown {

left: 100%;

top: 0;

}

.top-bar-section > ul > .divider, .top-bar-section > ul > [role="separator"] {

border-bottom: none;

border-top: none;

border-right: solid 1px #2b2b2b;

border-left: solid 1px black;

clear: none;

height: 45px;

width: 0px;

}

.top-bar-section .has-form {

background: #111111;

padding: 0 15px;

height: 45px;

}

.top-bar-section ul.right li .dropdown {

left: auto;

right: 0;

}

.top-bar-section ul.right li .dropdown li .dropdown {

right: 100%;

}

.no-js .top-bar-section ul li:hover > a {

background: #090909;

color: white;

}

.no-js .top-bar-section ul li:active > a {

background: #090909;

color: white;

}

.no-js .top-bar-section .has-dropdown:hover > .dropdown {

visibility: visible;

}

}

@-webkit-keyframes rotate {

from {

-webkit-transform: rotate(0deg);

}

to {

-webkit-transform: rotate(360deg);

}

}

@-moz-keyframes rotate {

from {

-webkit-transform: rotate(0deg);

}

to {

-webkit-transform: rotate(360deg);

}

}

@-o-keyframes rotate {

from {

-webkit-transform: rotate(0deg);

}

to {

-webkit-transform: rotate(360deg);

}

}

@keyframes rotate {

from {

-webkit-transform: rotate(0deg);

}

to {

-webkit-transform: rotate(360deg);

}

}

/\* Orbit Graceful Loading \*/

.slideshow-wrapper {

position: relative;

}

.slideshow-wrapper ul {

list-style-type: none;

margin: 0;

}

.slideshow-wrapper ul li,

.slideshow-wrapper ul li .orbit-caption {

display: none;

}

.slideshow-wrapper ul li:first-child {

display: block;

}

.slideshow-wrapper .orbit-container {

background-color: transparent;

}

.slideshow-wrapper .orbit-container li {

display: block;

}

.slideshow-wrapper .orbit-container li .orbit-caption {

display: block;

}

.preloader {

display: block;

width: 40px;

height: 40px;

position: absolute;

top: 50%;

left: 50%;

margin-top: -20px;

margin-left: -20px;

border: solid 3px;

border-color: #555555 white;

-webkit-border-radius: 1000px;

border-radius: 1000px;

-webkit-animation-name: rotate;

-webkit-animation-duration: 1.5s;

-webkit-animation-iteration-count: infinite;

-webkit-animation-timing-function: linear;

-moz-animation-name: rotate;

-moz-animation-duration: 1.5s;

-moz-animation-iteration-count: infinite;

-moz-animation-timing-function: linear;

-o-animation-name: rotate;

-o-animation-duration: 1.5s;

-o-animation-iteration-count: infinite;

-o-animation-timing-function: linear;

animation-name: rotate;

animation-duration: 1.5s;

animation-iteration-count: infinite;

animation-timing-function: linear;

}

.orbit-container {

overflow: hidden;

width: 100%;

position: relative;

background: whitesmoke;

}

.orbit-container .orbit-slides-container {

list-style: none;

margin: 0;

padding: 0;

position: relative;

}

.orbit-container .orbit-slides-container img {

display: block;

max-width: 100%;

}

.orbit-container .orbit-slides-container > \* {

position: absolute;

top: 0;

width: 100%;

margin-left: 100%;

}

.orbit-container .orbit-slides-container > \*:first-child {

margin-left: 0%;

}

.orbit-container .orbit-slides-container > \* .orbit-caption {

position: absolute;

bottom: 0;

background-color: black;

background-color: rgba(0, 0, 0, 0.6);

color: white;

width: 100%;

padding: 10px 14px;

font-size: 0.875em;

}

.orbit-container .orbit-slide-number {

position: absolute;

top: 10px;

left: 10px;

font-size: 12px;

color: white;

background: rgba(0, 0, 0, 0);

z-index: 10;

}

.orbit-container .orbit-slide-number span {

font-weight: 700;

padding: 0.3125em;

}

.orbit-container .orbit-timer {

position: absolute;

top: 10px;

right: 10px;

height: 6px;

width: 100px;

z-index: 10;

}

.orbit-container .orbit-timer .orbit-progress {

height: 100%;

background-color: black;

background-color: rgba(0, 0, 0, 0.6);

display: block;

width: 0%;

}

.orbit-container .orbit-timer > span {

display: none;

position: absolute;

top: 10px;

right: 0px;

width: 11px;

height: 14px;

border: solid 4px black;

border-top: none;

border-bottom: none;

}

.orbit-container .orbit-timer.paused > span {

right: -6px;

top: 9px;

width: 11px;

height: 14px;

border: inset 8px;

border-right-style: solid;

border-color: transparent transparent transparent black;

}

.orbit-container:hover .orbit-timer > span {

display: block;

}

.orbit-container .orbit-prev,

.orbit-container .orbit-next {

position: absolute;

top: 50%;

margin-top: -25px;

background-color: black;

background-color: rgba(0, 0, 0, 0.6);

width: 50px;

height: 60px;

line-height: 50px;

color: white;

text-indent: -9999px !important;

z-index: 10;

}

.orbit-container .orbit-prev > span,

.orbit-container .orbit-next > span {

position: absolute;

top: 50%;

margin-top: -16px;

display: block;

width: 0;

height: 0;

border: inset 16px;

}

.orbit-container .orbit-prev {

left: 0;

}

.orbit-container .orbit-prev > span {

border-right-style: solid;

border-color: transparent;

border-right-color: #fff;

}

.orbit-container .orbit-prev:hover > span {

border-right-color: #ccc;

}

.orbit-container .orbit-next {

right: 0;

}

.orbit-container .orbit-next > span {

border-color: transparent;

border-left-style: solid;

border-left-color: #fff;

left: 50%;

margin-left: -8px;

}

.orbit-container .orbit-next:hover > span {

border-left-color: #ccc;

}

.orbit-bullets {

margin: 0 auto 30px auto;

overflow: hidden;

position: relative;

top: 10px;

}

.orbit-bullets li {

display: block;

width: 10px;

height: 10px;

background: #999999;

float: left;

margin-right: 6px;

border: solid 1px #555555;

-webkit-border-radius: 1000px;

border-radius: 1000px;

}

.orbit-bullets li.active {

background: #555555;

}

.orbit-bullets li:last-child {

margin-right: 0;

}

.touch .orbit-container .orbit-prev,

.touch .orbit-container .orbit-next {

display: none;

}

.touch .orbit-bullets {

display: none;

}

@media only screen and (min-width: 768px) {

.touch .orbit-container .orbit-prev,

.touch .orbit-container .orbit-next {

display: inherit;

}

.touch .orbit-bullets {

display: block;

}

}

@media only screen and (max-width: 768px) {

.orbit-stack-on-small .orbit-slides-container {

height: auto !important;

}

.orbit-stack-on-small .orbit-slides-container > \* {

position: relative;

margin-left: 0% !important;

}

.orbit-stack-on-small .orbit-timer,

.orbit-stack-on-small .orbit-next,

.orbit-stack-on-small .orbit-prev,

.orbit-stack-on-small .orbit-bullets {

display: none;

}

}

.reveal-modal-bg {

position: fixed;

height: 100%;

width: 100%;

background: black;

background: rgba(0, 0, 0, 0.45);

z-index: 98;

display: none;

top: 0;

left: 0;

}

.reveal-modal {

visibility: hidden;

display: none;

position: absolute;

left: 50%;

z-index: 99;

height: auto;

margin-left: -40%;

width: 80%;

background-color: white;

padding: 1.25em;

border: solid 1px #666666;

-webkit-box-shadow: 0 0 10px rgba(0, 0, 0, 0.4);

box-shadow: 0 0 10px rgba(0, 0, 0, 0.4);

top: 50px;

}

.reveal-modal .column,

.reveal-modal .columns {

min-width: 0;

}

.reveal-modal > :first-child {

margin-top: 0;

}

.reveal-modal > :last-child {

margin-bottom: 0;

}

.reveal-modal .close-reveal-modal {

font-size: 1.375em;

line-height: 1;

position: absolute;

top: 0.5em;

right: 0.6875em;

color: #aaaaaa;

font-weight: bold;

cursor: pointer;

}

@media only screen and (min-width: 768px) {

.reveal-modal {

padding: 1.875em;

top: 6.25em;

}

.reveal-modal.tiny {

margin-left: -15%;

width: 30%;

}

.reveal-modal.small {

margin-left: -20%;

width: 40%;

}

.reveal-modal.medium {

margin-left: -30%;

width: 60%;

}

.reveal-modal.large {

margin-left: -35%;

width: 70%;

}

.reveal-modal.xlarge {

margin-left: -47.5%;

width: 95%;

}

}

@media print {

.reveal-modal {

background: white !important;

}

}

/\* Foundation Joyride \*/

.joyride-list {

display: none;

}

/\* Default styles for the container \*/

.joyride-tip-guide {

display: none;

position: absolute;

background: black;

color: white;

z-index: 101;

top: 0;

left: 2.5%;

font-family: inherit;

font-weight: normal;

width: 95%;

}

.lt-ie9 .joyride-tip-guide {

max-width: 800px;

left: 50%;

margin-left: -400px;

}

.joyride-content-wrapper {

width: 100%;

padding: 1.125em 1.25em 1.5em;

}

.joyride-content-wrapper .button {

margin-bottom: 0 !important;

}

/\* Add a little css triangle pip, older browser just miss out on the fanciness of it \*/

.joyride-tip-guide .joyride-nub {

display: block;

position: absolute;

left: 22px;

width: 0;

height: 0;

border: inset 14px;

}

.joyride-tip-guide .joyride-nub.top {

border-top-style: solid;

border-color: black;

border-top-color: transparent !important;

border-left-color: transparent !important;

border-right-color: transparent !important;

top: -28px;

}

.joyride-tip-guide .joyride-nub.bottom {

border-bottom-style: solid;

border-color: black !important;

border-bottom-color: transparent !important;

border-left-color: transparent !important;

border-right-color: transparent !important;

bottom: -28px;

}

.joyride-tip-guide .joyride-nub.right {

right: -28px;

}

.joyride-tip-guide .joyride-nub.left {

left: -28px;

}

/\* Typography \*/

.joyride-tip-guide h1,

.joyride-tip-guide h2,

.joyride-tip-guide h3,

.joyride-tip-guide h4,

.joyride-tip-guide h5,

.joyride-tip-guide h6 {

line-height: 1.25;

margin: 0;

font-weight: bold;

color: white;

}

.joyride-tip-guide p {

margin: 0 0 1.125em 0;

font-size: 0.875em;

line-height: 1.3;

}

.joyride-timer-indicator-wrap {

width: 50px;

height: 3px;

border: solid 1px #555555;

position: absolute;

right: 1.0625em;

bottom: 1em;

}

.joyride-timer-indicator {

display: block;

width: 0;

height: inherit;

background: #666666;

}

.joyride-close-tip {

position: absolute;

right: 12px;

top: 10px;

color: #777777 !important;

text-decoration: none;

font-size: 30px;

font-weight: normal;

line-height: 0.5 !important;

}

.joyride-close-tip:hover, .joyride-close-tip:focus {

color: #eeeeee !important;

}

.joyride-modal-bg {

position: fixed;

height: 100%;

width: 100%;

background: transparent;

background: rgba(0, 0, 0, 0.5);

z-index: 100;

display: none;

top: 0;

left: 0;

cursor: pointer;

}

.joyride-expose-wrapper {

background-color: #ffffff;

position: absolute;

border-radius: 3px;

z-index: 102;

-moz-box-shadow: 0px 0px 30px white;

-webkit-box-shadow: 0px 0px 15px white;

box-shadow: 0px 0px 15px white;

}

.joyride-expose-cover {

background: transparent;

border-radius: 3px;

position: absolute;

z-index: 9999;

top: 0px;

left: 0px;

}

/\* Styles for screens that are atleast 768px; \*/

@media only screen and (min-width: 768px) {

.joyride-tip-guide {

width: 300px;

left: inherit;

}

.joyride-tip-guide .joyride-nub.bottom {

border-color: black !important;

border-bottom-color: transparent !important;

border-left-color: transparent !important;

border-right-color: transparent !important;

bottom: -28px;

}

.joyride-tip-guide .joyride-nub.right {

border-color: black !important;

border-top-color: transparent !important;

border-right-color: transparent !important;

border-bottom-color: transparent !important;

top: 22px;

left: auto;

right: -28px;

}

.joyride-tip-guide .joyride-nub.left {

border-color: black !important;

border-top-color: transparent !important;

border-left-color: transparent !important;

border-bottom-color: transparent !important;

top: 22px;

left: -28px;

right: auto;

}

}

/\* Clearing Styles \*/

[data-clearing] {

\*zoom: 1;

margin-bottom: 0;

margin-left: 0;

list-style: none;

}

[data-clearing]:before, [data-clearing]:after {

content: " ";

display: table;

}

[data-clearing]:after {

clear: both;

}

[data-clearing] li {

float: left;

margin-right: 10px;

}

.clearing-blackout {

background: #111111;

position: fixed;

width: 100%;

height: 100%;

top: 0;

left: 0;

z-index: 998;

}

.clearing-blackout .clearing-close {

display: block;

}

.clearing-container {

position: relative;

z-index: 998;

height: 100%;

overflow: hidden;

margin: 0;

}

.visible-img {

height: 95%;

position: relative;

}

.visible-img img {

position: absolute;

left: 50%;

top: 50%;

margin-left: -50%;

max-height: 100%;

max-width: 100%;

}

.clearing-caption {

color: white;

line-height: 1.3;

margin-bottom: 0;

text-align: center;

bottom: 0;

background: #111111;

width: 100%;

padding: 10px 30px;

position: absolute;

left: 0;

}

.clearing-close {

z-index: 999;

padding-left: 20px;

padding-top: 10px;

font-size: 40px;

line-height: 1;

color: white;

display: none;

}

.clearing-close:hover, .clearing-close:focus {

color: #ccc;

}

.clearing-assembled .clearing-container {

height: 100%;

}

.clearing-assembled .clearing-container .carousel > ul {

display: none;

}

.clearing-feature li {

display: none;

}

.clearing-feature li.clearing-featured-img {

display: block;

}

@media only screen and (min-width: 768px) {

.clearing-main-prev,

.clearing-main-next {

position: absolute;

height: 100%;

width: 40px;

top: 0;

}

.clearing-main-prev > span,

.clearing-main-next > span {

position: absolute;

top: 50%;

display: block;

width: 0;

height: 0;

border: solid 16px;

}

.clearing-main-prev {

left: 0;

}

.clearing-main-prev > span {

left: 5px;

border-color: transparent;

border-right-color: white;

}

.clearing-main-next {

right: 0;

}

.clearing-main-next > span {

border-color: transparent;

border-left-color: white;

}

.clearing-main-prev.disabled,

.clearing-main-next.disabled {

opacity: 0.5;

}

.clearing-assembled .clearing-container .carousel {

background: #111111;

height: 150px;

margin-top: 5px;

}

.clearing-assembled .clearing-container .carousel > ul {

display: block;

z-index: 999;

width: 200%;

height: 100%;

margin-left: 0;

position: relative;

left: 0;

}

.clearing-assembled .clearing-container .carousel > ul li {

display: block;

width: 175px;

height: inherit;

padding: 0;

float: left;

overflow: hidden;

margin-right: 1px;

position: relative;

cursor: pointer;

opacity: 0.4;

}

.clearing-assembled .clearing-container .carousel > ul li.fix-height img {

min-height: 100%;

height: 100%;

max-width: none;

}

.clearing-assembled .clearing-container .carousel > ul li a.th {

border: none;

-webkit-box-shadow: none;

box-shadow: none;

display: block;

}

.clearing-assembled .clearing-container .carousel > ul li img {

cursor: pointer !important;

min-width: 100% !important;

}

.clearing-assembled .clearing-container .carousel > ul li.visible {

opacity: 1;

}

.clearing-assembled .clearing-container .visible-img {

background: #111111;

overflow: hidden;

height: 75%;

}

.clearing-close {

position: absolute;

top: 10px;

right: 20px;

padding-left: 0;

padding-top: 0;

}

}

/\* Foundation Alerts \*/

.alert-box {

border-style: solid;

border-width: 1px;

display: block;

font-weight: bold;

margin-bottom: 1.25em;

position: relative;

padding: 0.6875em 1.3125em 0.75em 0.6875em;

font-size: 0.875em;

background-color: #2ba6cb;

border-color: #2284a1;

color: white;

}

.alert-box .close {

font-size: 1.375em;

padding: 5px 4px 4px;

line-height: 0;

position: absolute;

top: 0.4375em;

right: 0.3125em;

color: #333333;

opacity: 0.3;

}

.alert-box .close:hover, .alert-box .close:focus {

opacity: 0.5;

}

.alert-box.radius {

-webkit-border-radius: 3px;

border-radius: 3px;

}

.alert-box.round {

-webkit-border-radius: 1000px;

border-radius: 1000px;

}

.alert-box.success {

background-color: #5da423;

border-color: #457a1a;

color: white;

}

.alert-box.alert {

background-color: #c60f13;

border-color: #970b0e;

color: white;

}

.alert-box.secondary {

background-color: #e9e9e9;

border-color: #d0d0d0;

color: #505050;

}

/\* Breadcrumbs \*/

.breadcrumbs {

display: block;

padding: 0.5625em 0.875em 0.5625em;

overflow: hidden;

margin-left: 0;

list-style: none;

border-style: solid;

border-width: 1px;

background-color: #f6f6f6;

border-color: gainsboro;

-webkit-border-radius: 3px;

border-radius: 3px;

}

.breadcrumbs > \* {

margin: 0;

float: left;

font-size: 0.6875em;

text-transform: uppercase;

}

.breadcrumbs > \*:hover a, .breadcrumbs > \*:focus a {

text-decoration: underline;

}

.breadcrumbs > \* a,

.breadcrumbs > \* span {

text-transform: uppercase;

color: #2ba6cb;

}

.breadcrumbs > \*.current {

cursor: default;

color: #333333;

}

.breadcrumbs > \*.current a {

cursor: default;

color: #333333;

}

.breadcrumbs > \*.current:hover, .breadcrumbs > \*.current:hover a, .breadcrumbs > \*.current:focus, .breadcrumbs > \*.current:focus a {

text-decoration: none;

}

.breadcrumbs > \*.unavailable {

color: #999999;

}

.breadcrumbs > \*.unavailable a {

color: #999999;

}

.breadcrumbs > \*.unavailable:hover, .breadcrumbs > \*.unavailable:hover a, .breadcrumbs > \*.unavailable:focus,

.breadcrumbs > \*.unavailable a:focus {

text-decoration: none;

color: #999999;

cursor: default;

}

.breadcrumbs > \*:before {

content: "/";

color: #aaaaaa;

margin: 0 0.75em;

position: relative;

top: 1px;

}

.breadcrumbs > \*:first-child:before {

content: " ";

margin: 0;

}

/\* Custom Checkbox and Radio Inputs \*/

form.custom .hidden-field {

margin-left: -99999px;

position: absolute;

visibility: hidden;

}

form.custom .custom {

display: inline-block;

width: 16px;

height: 16px;

position: relative;

top: -1px;

/\* fix centering issue \*/

vertical-align: middle;

border: solid 1px #cccccc;

background: white;

}

form.custom .custom.checkbox {

-webkit-border-radius: 0px;

border-radius: 0px;

padding: -1px;

}

form.custom .custom.radio {

-webkit-border-radius: 1000px;

border-radius: 1000px;

padding: 3px;

}

form.custom .custom.checkbox:before {

content: "";

display: block;

font-size: 16px;

color: white;

}

form.custom .custom.radio.checked:before {

content: "";

display: block;

width: 8px;

height: 8px;

-webkit-border-radius: 1000px;

border-radius: 1000px;

background: #222222;

position: relative;

}

form.custom .custom.checkbox.checked:before {

content: "\00d7";

color: #222222;

position: absolute;

top: -50%;

left: 50%;

margin-top: 4px;

margin-left: -5px;

}

/\* Custom Select Options and Dropdowns \*/

form.custom {

/\* Custom input, disabled \*/

}

form.custom .custom.dropdown {

display: block;

position: relative;

top: 0;

height: 2.3125em;

margin-bottom: 1.25em;

margin-top: 0px;

padding: 0px;

width: 100%;

background: white;

background: -moz-linear-gradient(top, white 0%, #f3f3f3 100%);

background: -webkit-linear-gradient(top, white 0%, #f3f3f3 100%);

-webkit-box-shadow: none;

background: linear-gradient(to bottom, white 0%, #f3f3f3 100%);

box-shadow: none;

font-size: 0.875em;

vertical-align: top;

}

form.custom .custom.dropdown ul {

overflow-y: auto;

max-height: 200px;

}

form.custom .custom.dropdown .current {

cursor: default;

white-space: nowrap;

line-height: 2.25em;

color: rgba(0, 0, 0, 0.75);

text-decoration: none;

overflow: hidden;

display: block;

margin-left: 0.5em;

margin-right: 2.3125em;

}

form.custom .custom.dropdown .selector {

cursor: default;

position: absolute;

width: 2.5em;

height: 2.3125em;

display: block;

right: 0;

top: 0;

}

form.custom .custom.dropdown .selector:after {

content: "";

display: block;

content: "";

display: block;

width: 0;

height: 0;

border: inset 5px;

border-color: #aaaaaa transparent transparent transparent;

border-top-style: solid;

position: absolute;

left: 0.9375em;

top: 50%;

margin-top: -3px;

}

form.custom .custom.dropdown:hover a.selector:after, form.custom .custom.dropdown.open a.selector:after {

content: "";

display: block;

width: 0;

height: 0;

border: inset 5px;

border-color: #222222 transparent transparent transparent;

border-top-style: solid;

}

form.custom .custom.dropdown .disabled {

color: #888888;

}

form.custom .custom.dropdown .disabled:hover {

background: transparent;

color: #888888;

}

form.custom .custom.dropdown .disabled:hover:after {

display: none;

}

form.custom .custom.dropdown.open ul {

display: block;

z-index: 10;

min-width: 100%;

-moz-box-sizing: content-box;

-webkit-box-sizing: content-box;

box-sizing: content-box;

}

form.custom .custom.dropdown.small {

max-width: 134px;

}

form.custom .custom.dropdown.medium {

max-width: 254px;

}

form.custom .custom.dropdown.large {

max-width: 434px;

}

form.custom .custom.dropdown.expand {

width: 100% !important;

}

form.custom .custom.dropdown.open.small ul {

min-width: 134px;

-moz-box-sizing: border-box;

-webkit-box-sizing: border-box;

box-sizing: border-box;

}

form.custom .custom.dropdown.open.medium ul {

min-width: 254px;

-moz-box-sizing: border-box;

-webkit-box-sizing: border-box;

box-sizing: border-box;

}

form.custom .custom.dropdown.open.large ul {

min-width: 434px;

-moz-box-sizing: border-box;

-webkit-box-sizing: border-box;

box-sizing: border-box;

}

form.custom .error .custom.dropdown {

border-color: #c60f13;

background-color: rgba(198, 15, 19, 0.1);

background: rgba(198, 15, 19, 0.1);

margin-bottom: 0;

}

form.custom .error .custom.dropdown:focus {

background: #fafafa;

border-color: #999999;

}

form.custom .error .custom.dropdown + small.error {

margin-top: 0;

}

form.custom .custom.dropdown ul {

position: absolute;

width: auto;

display: none;

margin: 0;

left: -1px;

top: auto;

-webkit-box-shadow: 0 2px 2px 0px rgba(0, 0, 0, 0.1);

box-shadow: 0 2px 2px 0px rgba(0, 0, 0, 0.1);

margin: 0;

padding: 0;

background: white;

border: solid 1px #cccccc;

font-size: 16;

}

form.custom .custom.dropdown ul li {

color: #555555;

font-size: 0.875em;

cursor: default;

padding-top: 0.25em;

padding-bottom: 0.25em;

padding-left: 0.375em;

padding-right: 2.375em;

min-height: 1.5em;

line-height: 1.5em;

margin: 0;

white-space: nowrap;

list-style: none;

}

form.custom .custom.dropdown ul li.selected {

background: #eeeeee;

color: black;

}

form.custom .custom.dropdown ul li:hover {

background-color: #e4e4e4;

color: black;

}

form.custom .custom.dropdown ul li.selected:hover {

background: #eeeeee;

cursor: default;

color: black;

}

form.custom .custom.dropdown ul.show {

display: block;

}

form.custom .custom.disabled {

background: #dddddd;

}

/\* Keystroke Characters \*/

.keystroke,

kbd {

background-color: #ededed;

border-color: #dbdbdb;

color: #222222;

border-style: solid;

border-width: 1px;

margin: 0;

font-family: "Consolas", "Menlo", "Courier", monospace;

font-size: 0.875em;

padding: 0.125em 0.25em 0;

-webkit-border-radius: 3px;

border-radius: 3px;

}

/\* Labels \*/

.label {

font-weight: bold;

text-align: center;

text-decoration: none;

line-height: 1;

white-space: nowrap;

display: inline-block;

position: relative;

padding: 0.1875em 0.625em 0.25em;

font-size: 0.875em;

background-color: #2ba6cb;

color: white;

}

.label.radius {

-webkit-border-radius: 3px;

border-radius: 3px;

}

.label.round {

-webkit-border-radius: 1000px;

border-radius: 1000px;

}

.label.alert {

background-color: #c60f13;

color: white;

}

.label.success {

background-color: #5da423;

color: white;

}

.label.secondary {

background-color: #e9e9e9;

color: #333333;

}

/\* Inline Lists \*/

.inline-list {

margin: 0 auto 1.0625em auto;

margin-left: -1.375em;

margin-right: 0;

padding: 0;

list-style: none;

overflow: hidden;

}

.inline-list > li {

list-style: none;

float: left;

margin-left: 1.375em;

display: block;

}

.inline-list > li > \* {

display: block;

}

/\* Default Pagination \*/

ul.pagination {

display: block;

height: 1.5em;

margin-left: -0.3125em;

}

ul.pagination li {

height: 1.5em;

color: #222222;

font-size: 0.875em;

margin-left: 0.3125em;

}

ul.pagination li a {

display: block;

padding: 0.0625em 0.4375em 0.0625em;

color: #999999;

}

ul.pagination li:hover a,

ul.pagination li a:focus {

background: #e6e6e6;

}

ul.pagination li.unavailable a {

cursor: default;

color: #999999;

}

ul.pagination li.unavailable:hover a, ul.pagination li.unavailable a:focus {

background: transparent;

}

ul.pagination li.current a {

background: #2ba6cb;

color: white;

font-weight: bold;

cursor: default;

}

ul.pagination li.current a:hover, ul.pagination li.current a:focus {

background: #2ba6cb;

}

ul.pagination li {

float: left;

display: block;

}

/\* Pagination centred wrapper \*/

.pagination-centered {

text-align: center;

}

.pagination-centered ul.pagination li {

float: none;

display: inline-block;

}

/\* Panels \*/

.panel {

/\*border-style: solid;

border-width: 1px;

border-color: #d9d9d9;

margin-bottom: 1.25em; \*/

padding: 1.25em;

background: #f2f2f2;

}

.panel > :first-child {

margin-top: 0;

}

.panel > :last-child {

margin-bottom: 0;

}

.panel h1, .panel h2, .panel h3, .panel h4, .panel h5, .panel h6, .panel p {

color: #333333;

}

.panel h1, .panel h2, .panel h3, .panel h4, .panel h5, .panel h6 {

line-height: 1;

margin-bottom: 0.625em;

}

.panel h1.subheader, .panel h2.subheader, .panel h3.subheader, .panel h4.subheader, .panel h5.subheader, .panel h6.subheader {

line-height: 1.4;

}

.panel.callout {

border-style: solid;

border-width: 1px;

border-color: #2284a1;

margin-bottom: 1.25em;

padding: 1.25em;

background: #2ba6cb;

-webkit-box-shadow: 0 1px 0 rgba(255, 255, 255, 0.5) inset;

box-shadow: 0 1px 0 rgba(255, 255, 255, 0.5) inset;

}

.panel.callout > :first-child {

margin-top: 0;

}

.panel.callout > :last-child {

margin-bottom: 0;

}

.panel.callout h1, .panel.callout h2, .panel.callout h3, .panel.callout h4, .panel.callout h5, .panel.callout h6, .panel.callout p {

color: white;

}

.panel.callout h1, .panel.callout h2, .panel.callout h3, .panel.callout h4, .panel.callout h5, .panel.callout h6 {

line-height: 1;

margin-bottom: 0.625em;

}

.panel.callout h1.subheader, .panel.callout h2.subheader, .panel.callout h3.subheader, .panel.callout h4.subheader, .panel.callout h5.subheader, .panel.callout h6.subheader {

line-height: 1.4;

}

.panel.radius {

-webkit-border-radius: 3px;

border-radius: 3px;

}

/\* Pricing Tables \*/

.pricing-table {

border: solid 1px #dddddd;

margin-left: 0;

margin-bottom: 1.25em;

}

.pricing-table \* {

list-style: none;

line-height: 1;

}

.pricing-table .title {

background-color: #dddddd;

padding: 0.9375em 1.25em;

text-align: center;

color: #333333;

font-weight: bold;

font-size: 1em;

}

.pricing-table .price {

background-color: #eeeeee;

padding: 0.9375em 1.25em;

text-align: center;

color: #333333;

font-weight: normal;

font-size: 1.25em;

}

.pricing-table .description {

background-color: white;

padding: 0.9375em;

text-align: center;

color: #777777;

font-size: 0.75em;

font-weight: normal;

line-height: 1.4;

border-bottom: dotted 1px #dddddd;

}

.pricing-table .bullet-item {

background-color: white;

padding: 0.9375em;

text-align: center;

color: #333333;

font-size: 0.875em;

font-weight: normal;

border-bottom: dotted 1px #dddddd;

}

.pricing-table .cta-button {

background-color: whitesmoke;

text-align: center;

padding: 1.25em 1.25em 0;

}

/\* Progress Bar \*/

.progress {

background-color: transparent;

height: 1.5625em;

border: 1px solid #cccccc;

padding: 0.125em;

margin-bottom: 0.625em;

}

.progress .meter {

background: #2ba6cb;

height: 100%;

display: block;

}

.progress.secondary .meter {

background: #e9e9e9;

height: 100%;

display: block;

}

.progress.success .meter {

background: #5da423;

height: 100%;

display: block;

}

.progress.alert .meter {

background: #c60f13;

height: 100%;

display: block;

}

.progress.radius {

-webkit-border-radius: 3px;

border-radius: 3px;

}

.progress.radius .meter {

-webkit-border-radius: 2px;

border-radius: 2px;

}

.progress.round {

-webkit-border-radius: 1000px;

border-radius: 1000px;

}

.progress.round .meter {

-webkit-border-radius: 999px;

border-radius: 999px;

}

/\* Side Nav \*/

.side-nav {

display: block;

margin: 0;

padding: 0.875em 0;

list-style-type: none;

list-style-position: inside;

}

.side-nav li {

margin: 0 0 0.4375em 0;

font-size: 0.875em;

}

.side-nav li a {

display: block;

color: #2ba6cb;

}

.side-nav li.active > a:first-child {

color: #4d4d4d;

font-weight: bold;

}

.side-nav li.divider {

border-top: 1px solid;

height: 0;

padding: 0;

list-style: none;

border-top-color: #e6e6e6;

}

/\* Side Nav \*/

.sub-nav {

display: block;

width: auto;

overflow: hidden;

margin: -0.25em 0 1.125em;

padding-top: 0.25em;

margin-right: 0;

margin-left: -0.5625em;

}

.sub-nav dt,

.sub-nav dd {

float: left;

display: inline;

margin-left: 0.5625em;

margin-bottom: 0.625em;

font-weight: normal;

font-size: 0.875em;

}

.sub-nav dt a,

.sub-nav dd a {

color: #999999;

text-decoration: none;

}

.sub-nav dt.active a,

.sub-nav dd.active a {

-webkit-border-radius: 1000px;

border-radius: 1000px;

font-weight: bold;

background: #2ba6cb;

padding: 0.1875em 0.5625em;

cursor: default;

color: white;

}

/\* Foundation Switches \*/

@media only screen {

div.switch {

position: relative;

width: 100%;

padding: 0;

display: block;

overflow: hidden;

border-style: solid;

border-width: 1px;

margin-bottom: 1.25em;

-webkit-animation: webkitSiblingBugfix infinite 1s;

height: 36px;

background: white;

border-color: #cccccc;

}

div.switch label {

position: relative;

left: 0;

z-index: 2;

float: left;

width: 50%;

height: 100%;

margin: 0;

font-weight: bold;

text-align: left;

-webkit-transition: all 0.1s ease-out;

-moz-transition: all 0.1s ease-out;

transition: all 0.1s ease-out;

}

div.switch input {

position: absolute;

z-index: 3;

opacity: 0;

width: 100%;

height: 100%;

-moz-appearance: none;

}

div.switch input:hover, div.switch input:focus {

cursor: pointer;

}

div.switch span:last-child {

position: absolute;

top: -1px;

left: -1px;

z-index: 1;

display: block;

padding: 0;

border-width: 1px;

border-style: solid;

-webkit-transition: all 0.1s ease-out;

-moz-transition: all 0.1s ease-out;

transition: all 0.1s ease-out;

}

div.switch input:not(:checked) + label {

opacity: 0;

}

div.switch input:checked {

display: none !important;

}

div.switch input {

left: 0;

display: block !important;

}

div.switch input:first-of-type + label,

div.switch input:first-of-type + span + label {

left: -50%;

}

div.switch input:first-of-type:checked + label,

div.switch input:first-of-type:checked + span + label {

left: 0%;

}

div.switch input:last-of-type + label,

div.switch input:last-of-type + span + label {

right: -50%;

left: auto;

text-align: right;

}

div.switch input:last-of-type:checked + label,

div.switch input:last-of-type:checked + span + label {

right: 0%;

left: auto;

}

div.switch span.custom {

display: none !important;

}

form.custom div.switch .hidden-field {

margin-left: auto;

position: absolute;

visibility: visible;

}

div.switch label {

padding: 0 0.375em;

line-height: 2.3em;

font-size: 0.875em;

}

div.switch input:first-of-type:checked ~ span:last-child {

left: 100%;

margin-left: -2.1875em;

}

div.switch span:last-child {

width: 2.25em;

height: 2.25em;

}

div.switch span:last-child {

border-color: #b3b3b3;

background: white;

background: -moz-linear-gradient(top, white 0%, #f2f2f2 100%);

background: -webkit-linear-gradient(top, white 0%, #f2f2f2 100%);

background: linear-gradient(to bottom, white 0%, #f2f2f2 100%);

-webkit-box-shadow: 2px 0 10px 0 rgba(0, 0, 0, 0.07), 1000px 0 0 1000px #e1f5d1, -2px 0 10px 0 rgba(0, 0, 0, 0.07), -1000px 0 0 1000px whitesmoke;

box-shadow: 2px 0 10px 0 rgba(0, 0, 0, 0.07), 1000px 0 0 980px #e1f5d1, -2px 0 10px 0 rgba(0, 0, 0, 0.07), -1000px 0 0 1000px whitesmoke;

}

div.switch:hover span:last-child, div.switch:focus span:last-child {

background: white;

background: -moz-linear-gradient(top, white 0%, #e6e6e6 100%);

background: -webkit-linear-gradient(top, white 0%, #e6e6e6 100%);

background: linear-gradient(to bottom, white 0%, #e6e6e6 100%);

}

div.switch:active {

background: transparent;

}

div.switch.large {

height: 44px;

}

div.switch.large label {

padding: 0 0.375em;

line-height: 2.3em;

font-size: 1.0625em;

}

div.switch.large input:first-of-type:checked ~ span:last-child {

left: 100%;

margin-left: -2.6875em;

}

div.switch.large span:last-child {

width: 2.75em;

height: 2.75em;

}

div.switch.small {

height: 28px;

}

div.switch.small label {

padding: 0 0.375em;

line-height: 2.1em;

font-size: 0.75em;

}

div.switch.small input:first-of-type:checked ~ span:last-child {

left: 100%;

margin-left: -1.6875em;

}

div.switch.small span:last-child {

width: 1.75em;

height: 1.75em;

}

div.switch.tiny {

height: 22px;

}

div.switch.tiny label {

padding: 0 0.375em;

line-height: 1.9em;

font-size: 0.6875em;

}

div.switch.tiny input:first-of-type:checked ~ span:last-child {

left: 100%;

margin-left: -1.3125em;

}

div.switch.tiny span:last-child {

width: 1.375em;

height: 1.375em;

}

div.switch.radius {

-webkit-border-radius: 4px;

border-radius: 4px;

}

div.switch.radius span:last-child {

-webkit-border-radius: 3px;

border-radius: 3px;

}

div.switch.round {

-webkit-border-radius: 1000px;

border-radius: 1000px;

}

div.switch.round span:last-child {

-webkit-border-radius: 999px;

border-radius: 999px;

}

div.switch.round label {

padding: 0 0.5625em;

}

@-webkit-keyframes webkitSiblingBugfix {

from {

position: relative;

}

to {

position: relative;

}

}

}

[data-magellan-expedition] {

background: white;

z-index: 50;

min-width: 100%;

padding: 10px;

}

[data-magellan-expedition] .sub-nav {

margin-bottom: 0;

}

[data-magellan-expedition] .sub-nav dd {

margin-bottom: 0;

}

/\* Tables \*/

table {

background: white;

margin-bottom: 1.25em;

border: solid 1px #dddddd;

}

table thead,

table tfoot {

background: whitesmoke;

font-weight: bold;

}

table thead tr th,

table thead tr td,

table tfoot tr th,

table tfoot tr td {

padding: 0.5em 0.625em 0.625em;

font-size: 0.875em;

color: #222222;

text-align: left;

}

table tr th,

table tr td {

padding: 0.5625em 0.625em;

font-size: 0.875em;

color: #222222;

}

table tr.even, table tr.alt, table tr:nth-of-type(even) {

background: #f9f9f9;

}

table thead tr th,

table tfoot tr th,

table tbody tr td,

table tr td,

table tfoot tr td {

display: table-cell;

line-height: 1.125em;

}

/\* Image Thumbnails \*/

.th {

line-height: 0;

display: inline-block;

border: solid 4px white;

-webkit-box-shadow: 0 0 0 1px rgba(0, 0, 0, 0.2);

box-shadow: 0 0 0 1px rgba(0, 0, 0, 0.2);

-webkit-transition: all 200ms ease-out;

-moz-transition: all 200ms ease-out;

transition: all 200ms ease-out;

}

.th:hover, .th:focus {

-webkit-box-shadow: 0 0 6px 1px rgba(43, 166, 203, 0.5);

box-shadow: 0 0 6px 1px rgba(43, 166, 203, 0.5);

}

.th.radius {

-webkit-border-radius: 3px;

border-radius: 3px;

}

a.th {

display: inline-block;

}

/\* Tooltips \*/

.has-tip {

border-bottom: dotted 1px #cccccc;

cursor: help;

font-weight: bold;

color: #333333;

}

.has-tip:hover, .has-tip:focus {

border-bottom: dotted 1px #196177;

color: #2ba6cb;

}

.has-tip.tip-left, .has-tip.tip-right {

float: none !important;

}

.tooltip {

display: none;

position: absolute;

z-index: 999;

font-weight: bold;

font-size: 0.9375em;

line-height: 1.3;

padding: 0.5em;

max-width: 85%;

left: 50%;

width: 100%;

color: white;

background: black;

-webkit-border-radius: 3px;

border-radius: 3px;

}

.tooltip > .nub {

display: block;

left: 5px;

position: absolute;

width: 0;

height: 0;

border: solid 5px;

border-color: transparent transparent black transparent;

top: -10px;

}

.tooltip.opened {

color: #2ba6cb !important;

border-bottom: dotted 1px #196177 !important;

}

.tap-to-close {

display: block;

font-size: 0.625em;

color: #888888;

font-weight: normal;

}

@media only screen and (min-width: 768px) {

.tooltip > .nub {

border-color: transparent transparent black transparent;

top: -10px;

}

.tooltip.tip-top > .nub {

border-color: black transparent transparent transparent;

top: auto;

bottom: -10px;

}

.tooltip.tip-left, .tooltip.tip-right {

float: none !important;

}

.tooltip.tip-left > .nub {

border-color: transparent transparent transparent black;

right: -10px;

left: auto;

top: 50%;

margin-top: -5px;

}

.tooltip.tip-right > .nub {

border-color: transparent black transparent transparent;

right: auto;

left: -10px;

top: 50%;

margin-top: -5px;

}

}

@media only screen and (max-width: 767px) {

.f-dropdown {

max-width: 100%;

left: 0;

}

}

/\* Foundation Dropdowns \*/

.f-dropdown {

position: absolute;

top: -9999px;

list-style: none;

margin-left: 0;

width: 100%;

max-height: none;

height: auto;

background: white;

border: solid 1px #cccccc;

font-size: 16;

z-index: 99;

margin-top: 2px;

max-width: 200px;

}

.f-dropdown > \*:first-child {

margin-top: 0;

}

.f-dropdown > \*:last-child {

margin-bottom: 0;

}

.f-dropdown:before {

content: "";

display: block;

width: 0;

height: 0;

border: inset 6px;

border-color: transparent transparent white transparent;

border-bottom-style: solid;

position: absolute;

top: -12px;

left: 10px;

z-index: 99;

}

.f-dropdown:after {

content: "";

display: block;

width: 0;

height: 0;

border: inset 7px;

border-color: transparent transparent #cccccc transparent;

border-bottom-style: solid;

position: absolute;

top: -14px;

left: 9px;

z-index: 98;

}

.f-dropdown.right:before {

left: auto;

right: 10px;

}

.f-dropdown.right:after {

left: auto;

right: 9px;

}

.f-dropdown li {

font-size: 0.875em;

cursor: pointer;

line-height: 1.125em;

margin: 0;

}

.f-dropdown li:hover, .f-dropdown li:focus {

background: #eeeeee;

}

.f-dropdown li a {

display: block;

padding: 0.3125em 0.625em;

color: #555555;

}

.f-dropdown.content {

position: absolute;

top: -9999px;

list-style: none;

margin-left: 0;

padding: 1.25em;

width: 100%;

height: auto;

max-height: none;

background: white;

border: solid 1px #cccccc;

font-size: 16;

z-index: 99;

max-width: 200px;

}

.f-dropdown.content > \*:first-child {

margin-top: 0;

}

.f-dropdown.content > \*:last-child {

margin-bottom: 0;

}

.f-dropdown.tiny {

max-width: 200px;

}

.f-dropdown.small {

max-width: 300px;

}

.f-dropdown.medium {

max-width: 500px;

}

.f-dropdown.large {

max-width: 800px;

}

var validate = function (obj, keys) {

for (key in obj) {

if (obj.hasOwnProperty(key)) {

if (keys.hasOwnProperty(key)) {

if (typeof obj[key] !== keys[key])

throw new Error(format(ERROR.INVALID\_TYPE, [typeof obj[key], key]));

} else {

var errorStr = "Unknown property, " + key + ". Valid properties are:";

for (key in keys)

if (keys.hasOwnProperty(key))

errorStr = errorStr + " " + key;

throw new Error(errorStr);

}

}

}

};

/\*\*

\* Return a new function which runs the user function bound

\* to a fixed scope.

\* @param {function} User function

\* @param {object} Function scope

\* @return {function} User function bound to another scope

\* @private

\*/

var scope = function (f, scope) {

return function () {

return f.apply(scope, arguments);

};

};

/\*\*

\* Unique message type identifiers, with associated

\* associated integer values.

\* @private

\*/

var ERROR = {

OK: {code: 0, text: "AMQJSC0000I OK."},

CONNECT\_TIMEOUT: {code: 1, text: "AMQJSC0001E Connect timed out."},

SUBSCRIBE\_TIMEOUT: {code: 2, text: "AMQJS0002E Subscribe timed out."},

UNSUBSCRIBE\_TIMEOUT: {code: 3, text: "AMQJS0003E Unsubscribe timed out."},

PING\_TIMEOUT: {code: 4, text: "AMQJS0004E Ping timed out."},

INTERNAL\_ERROR: {code: 5, text: "AMQJS0005E Internal error."},

CONNACK\_RETURNCODE: {code: 6, text: "AMQJS0006E Bad Connack return code:{0} {1}."},

SOCKET\_ERROR: {code: 7, text: "AMQJS0007E Socket error:{0}."},

SOCKET\_CLOSE: {code: 8, text: "AMQJS0008I Socket closed."},

MALFORMED\_UTF: {code: 9, text: "AMQJS0009E Malformed UTF data:{0} {1} {2}."},

UNSUPPORTED: {code: 10, text: "AMQJS0010E {0} is not supported by this browser."},

INVALID\_STATE: {code: 11, text: "AMQJS0011E Invalid state {0}."},

INVALID\_TYPE: {code: 12, text: "AMQJS0012E Invalid type {0} for {1}."},

INVALID\_ARGUMENT: {code: 13, text: "AMQJS0013E Invalid argument {0} for {1}."},

UNSUPPORTED\_OPERATION: {code: 14, text: "AMQJS0014E Unsupported operation."},

INVALID\_STORED\_DATA: {code: 15, text: "AMQJS0015E Invalid data in local storage key={0} value={1}."},

INVALID\_MQTT\_MESSAGE\_TYPE: {code: 16, text: "AMQJS0016E Invalid MQTT message type {0}."},

MALFORMED\_UNICODE: {code: 17, text: "AMQJS0017E Malformed Unicode string:{0} {1}."},

};

/\*\* CONNACK RC Meaning. \*/

var CONNACK\_RC = {

0: "Connection Accepted",

1: "Connection Refused: unacceptable protocol version",

2: "Connection Refused: identifier rejected",

3: "Connection Refused: server unavailable",

4: "Connection Refused: bad user name or password",

5: "Connection Refused: not authorized"

};

/\*\*

\* Format an error message text.

\* @private

\* @param {error} ERROR.KEY value above.

\* @param {substitutions} [array] substituted into the text.

\* @return the text with the substitutions made.

\*/

var format = function (error, substitutions) {

var text = error.text;

if (substitutions) {

for (var i = 0; i < substitutions.length; i++) {

field = "{" + i + "}";

start = text.indexOf(field);

if (start > 0) {

var part1 = text.substring(0, start);

var part2 = text.substring(start + field.length);

text = part1 + substitutions[i] + part2;

}

}

}

return text;

};

//MQTT protocol and version 6 M Q I s d p 3

var MqttProtoIdentifier = [0x00, 0x06, 0x4d, 0x51, 0x49, 0x73, 0x64, 0x70, 0x03];

/\*\*

\* @ignore

\* Construct an MQTT wire protocol message.

\* @param type MQTT packet type.

\* @param options optional wire message attributes.

\*

\* Optional properties

\*

\* messageIdentifier: message ID in the range [0..65535]

\* payloadMessage: Application Message - PUBLISH only

\* connectStrings: array of 0 or more Strings to be put into the CONNECT payload

\* topics: array of strings (SUBSCRIBE, UNSUBSCRIBE)

\* requestQoS: array of QoS values [0..2]

\*

\* "Flag" properties

\* cleanSession: true if present / false if absent (CONNECT)

\* willMessage: true if present / false if absent (CONNECT)

\* isRetained: true if present / false if absent (CONNECT)

\* userName: true if present / false if absent (CONNECT)

\* password: true if present / false if absent (CONNECT)

\* keepAliveInterval: integer [0..65535] (CONNECT)

\*

\* @private

\*/

var WireMessage = function (type, options) {

this.type = type;

for (name in options) {

if (options.hasOwnProperty(name)) {

this[name] = options[name];

}

}

};

WireMessage.prototype.encode = function () {

// Compute the first byte of the fixed header

var first = ((this.type & 0x0f) << 4);

/\*

\* Now calculate the length of the variable header + payload by adding up the lengths

\* of all the component parts

\*/

remLength = 0;

topicStrLength = new Array();

// if the message contains a messageIdentifier then we need two bytes for that

if (this.messageIdentifier != undefined)

remLength += 2;

switch (this.type) {

// If this a Connect then we need to include 12 bytes for its header

case MESSAGE\_TYPE.CONNECT:

remLength += MqttProtoIdentifier.length + 3;

remLength += UTF8Length(this.clientId) + 2;

if (this.willMessage != undefined) {

remLength += UTF8Length(this.willMessage.destinationName) + 2;

// Will message is always a string, sent as UTF-8 characters with a preceding length.

var willMessagePayloadBytes = this.willMessage.payloadBytes;

if (!(willMessagePayloadBytes instanceof Uint8Array))

willMessagePayloadBytes = new Uint8Array(payloadBytes);

remLength += willMessagePayloadBytes.byteLength + 2;

}

if (this.userName != undefined)

remLength += UTF8Length(this.userName) + 2;

if (this.password != undefined)

remLength += UTF8Length(this.password) + 2;

break;

// Subscribe, Unsubscribe can both contain topic strings

case MESSAGE\_TYPE.SUBSCRIBE:

first |= 0x02; // Qos = 1;

for (var i = 0; i < this.topics.length; i++) {

topicStrLength[i] = UTF8Length(this.topics[i]);

remLength += topicStrLength[i] + 2;

}

remLength += this.requestedQos.length; // 1 byte for each topic's Qos

// QoS on Subscribe only

break;

case MESSAGE\_TYPE.UNSUBSCRIBE:

first |= 0x02; // Qos = 1;

for (var i = 0; i < this.topics.length; i++) {

topicStrLength[i] = UTF8Length(this.topics[i]);

remLength += topicStrLength[i] + 2;

}

break;

case MESSAGE\_TYPE.PUBLISH:

if (this.payloadMessage.duplicate) first |= 0x08;

first = first |= (this.payloadMessage.qos << 1);

if (this.payloadMessage.retained) first |= 0x01;

destinationNameLength = UTF8Length(this.payloadMessage.destinationName);

remLength += destinationNameLength + 2;

var payloadBytes = this.payloadMessage.payloadBytes;

remLength += payloadBytes.byteLength;

if (payloadBytes instanceof ArrayBuffer)

payloadBytes = new Uint8Array(payloadBytes);

else if (!(payloadBytes instanceof Uint8Array))

payloadBytes = new Uint8Array(payloadBytes.buffer);

break;

case MESSAGE\_TYPE.DISCONNECT:

break;

default:

;

}

// Now we can allocate a buffer for the message

var mbi = encodeMBI(remLength); // Convert the length to MQTT MBI format

var pos = mbi.length + 1; // Offset of start of variable header

var buffer = new ArrayBuffer(remLength + pos);

var byteStream = new Uint8Array(buffer); // view it as a sequence of bytes

//Write the fixed header into the buffer

byteStream[0] = first;

byteStream.set(mbi, 1);

// If this is a PUBLISH then the variable header starts with a topic

if (this.type == MESSAGE\_TYPE.PUBLISH)

pos = writeString(this.payloadMessage.destinationName, destinationNameLength, byteStream, pos);

// If this is a CONNECT then the variable header contains the protocol name/version, flags and keepalive time

else if (this.type == MESSAGE\_TYPE.CONNECT) {

byteStream.set(MqttProtoIdentifier, pos);

pos += MqttProtoIdentifier.length;

var connectFlags = 0;

if (this.cleanSession)

connectFlags = 0x02;

if (this.willMessage != undefined) {

connectFlags |= 0x04;

connectFlags |= (this.willMessage.qos << 3);

if (this.willMessage.retained) {

connectFlags |= 0x20;

}

}

if (this.userName != undefined)

connectFlags |= 0x80;

if (this.password != undefined)

connectFlags |= 0x40;

byteStream[pos++] = connectFlags;

pos = writeUint16(this.keepAliveInterval, byteStream, pos);

}

// Output the messageIdentifier - if there is one

if (this.messageIdentifier != undefined)

pos = writeUint16(this.messageIdentifier, byteStream, pos);

switch (this.type) {

case MESSAGE\_TYPE.CONNECT:

pos = writeString(this.clientId, UTF8Length(this.clientId), byteStream, pos);

if (this.willMessage != undefined) {

pos = writeString(this.willMessage.destinationName, UTF8Length(this.willMessage.destinationName), byteStream, pos);

pos = writeUint16(willMessagePayloadBytes.byteLength, byteStream, pos);

byteStream.set(willMessagePayloadBytes, pos);

pos += willMessagePayloadBytes.byteLength;

}

if (this.userName != undefined)

pos = writeString(this.userName, UTF8Length(this.userName), byteStream, pos);

if (this.password != undefined)

pos = writeString(this.password, UTF8Length(this.password), byteStream, pos);

break;

case MESSAGE\_TYPE.PUBLISH:

// PUBLISH has a text or binary payload, if text do not add a 2 byte length field, just the UTF characters.

byteStream.set(payloadBytes, pos);

break;

// case MESSAGE\_TYPE.PUBREC:

// case MESSAGE\_TYPE.PUBREL:

// case MESSAGE\_TYPE.PUBCOMP:

// break;

case MESSAGE\_TYPE.SUBSCRIBE:

// SUBSCRIBE has a list of topic strings and request QoS

for (var i = 0; i < this.topics.length; i++) {

pos = writeString(this.topics[i], topicStrLength[i], byteStream, pos);

byteStream[pos++] = this.requestedQos[i];

}

break;

case MESSAGE\_TYPE.UNSUBSCRIBE:

// UNSUBSCRIBE has a list of topic strings

for (var i = 0; i < this.topics.length; i++)

pos = writeString(this.topics[i], topicStrLength[i], byteStream, pos);

break;

default:

// Do nothing.

}

return buffer;

}

function decodeMessage(input) {

//var msg = new Object(); // message to be constructed

var first = input[0];

var type = first >> 4;

var messageInfo = first &= 0x0f;

var pos = 1;

// Decode the remaining length (MBI format)

var digit;

var remLength = 0;

var multiplier = 1;

do {

digit = input[pos++];

remLength += ((digit & 0x7F) \* multiplier);

multiplier \*= 128;

} while ((digit & 0x80) != 0);

var wireMessage = new WireMessage(type);

switch (type) {

case MESSAGE\_TYPE.CONNACK:

wireMessage.topicNameCompressionResponse = input[pos++];

wireMessage.returnCode = input[pos++];

break;

case MESSAGE\_TYPE.PUBLISH:

var qos = (messageInfo >> 1) & 0x03;

var len = readUint16(input, pos);

pos += 2;

var topicName = parseUTF8(input, pos, len);

pos += len;

// If QoS 1 or 2 there will be a messageIdentifier

if (qos > 0) {

wireMessage.messageIdentifier = readUint16(input, pos);

pos += 2;

}

var message = new Messaging.Message(input.subarray(pos));

if ((messageInfo & 0x01) == 0x01)

message.retained = true;

if ((messageInfo & 0x08) == 0x08)

message.duplicate = true;

message.qos = qos;

message.destinationName = topicName;

wireMessage.payloadMessage = message;

break;

case MESSAGE\_TYPE.PUBACK:

case MESSAGE\_TYPE.PUBREC:

case MESSAGE\_TYPE.PUBREL:

case MESSAGE\_TYPE.PUBCOMP:

case MESSAGE\_TYPE.UNSUBACK:

wireMessage.messageIdentifier = readUint16(input, pos);

break;

case MESSAGE\_TYPE.SUBACK:

wireMessage.messageIdentifier = readUint16(input, pos);

pos += 2;

wireMessage.grantedQos = input.subarray(pos);

break;

default:

;

}

return wireMessage;

}

function writeUint16(input, buffer, offset) {

buffer[offset++] = input >> 8; //MSB

buffer[offset++] = input % 256; //LSB

return offset;

}

function writeString(input, utf8Length, buffer, offset) {

offset = writeUint16(utf8Length, buffer, offset);

stringToUTF8(input, buffer, offset);

return offset + utf8Length;

}

function readUint16(buffer, offset) {

return 256 \* buffer[offset] + buffer[offset + 1];

}

/\*\*

\* Encodes an MQTT Multi-Byte Integer

\* @private

\*/

function encodeMBI(number) {

var output = new Array(1);

var numBytes = 0;

do {

var digit = number % 128;

number = number >> 7;

if (number > 0) {

digit |= 0x80;

}

output[numBytes++] = digit;

} while ((number > 0) && (numBytes < 4));

return output;

}

/\*\*

\* Takes a String and calculates its length in bytes when encoded in UTF8.

\* @private

\*/

function UTF8Length(input) {

var output = 0;

for (var i = 0; i < input.length; i++) {

var charCode = input.charCodeAt(i);

if (charCode > 0x7FF) {

// Surrogate pair means its a 4 byte character

if (0xD800 <= charCode && charCode <= 0xDBFF) {

i++;

output++;

}

output += 3;

}

else if (charCode > 0x7F)

output += 2;

else

output++;

}

return output;

}

/\*\*

\* Takes a String and writes it into an array as UTF8 encoded bytes.

\* @private

\*/

function stringToUTF8(input, output, start) {

var pos = start;

for (var i = 0; i < input.length; i++) {

var charCode = input.charCodeAt(i);

// Check for a surrogate pair.

if (0xD800 <= charCode && charCode <= 0xDBFF) {

lowCharCode = input.charCodeAt(++i);

if (isNaN(lowCharCode)) {

throw new Error(format(ERROR.MALFORMED\_UNICODE, [charCode, lowCharCode]));

}

charCode = ((charCode - 0xD800) << 10) + (lowCharCode - 0xDC00) + 0x10000;

}

if (charCode <= 0x7F) {

output[pos++] = charCode;

} else if (charCode <= 0x7FF) {

output[pos++] = charCode >> 6 & 0x1F | 0xC0;

output[pos++] = charCode & 0x3F | 0x80;

} else if (charCode <= 0xFFFF) {

output[pos++] = charCode >> 12 & 0x0F | 0xE0;

output[pos++] = charCode >> 6 & 0x3F | 0x80;

output[pos++] = charCode & 0x3F | 0x80;

} else {

output[pos++] = charCode >> 18 & 0x07 | 0xF0;

output[pos++] = charCode >> 12 & 0x3F | 0x80;

output[pos++] = charCode >> 6 & 0x3F | 0x80;

output[pos++] = charCode & 0x3F | 0x80;

}

;

}

return output;

}

function parseUTF8(input, offset, length) {

var output = "";

var utf16;

var pos = offset;

while (pos < offset + length) {

var byte1 = input[pos++];

if (byte1 < 128)

utf16 = byte1;

else {

var byte2 = input[pos++] - 128;

if (byte2 < 0)

throw new Error(format(ERROR.MALFORMED\_UTF, [byte1.toString(16), byte2.toString(16), ""]));

if (byte1 < 0xE0) // 2 byte character

utf16 = 64 \* (byte1 - 0xC0) + byte2;

else {

var byte3 = input[pos++] - 128;

if (byte3 < 0)

throw new Error(format(ERROR.MALFORMED\_UTF, [byte1.toString(16), byte2.toString(16), byte3.toString(16)]));

if (byte1 < 0xF0) // 3 byte character

utf16 = 4096 \* (byte1 - 0xE0) + 64 \* byte2 + byte3;

else {

var byte4 = input[pos++] - 128;

if (byte4 < 0)

throw new Error(format(ERROR.MALFORMED\_UTF, [byte1.toString(16), byte2.toString(16), byte3.toString(16), byte4.toString(16)]));

if (byte1 < 0xF8) // 4 byte character

utf16 = 262144 \* (byte1 - 0xF0) + 4096 \* byte2 + 64 \* byte3 + byte4;

else // longer encodings are not supported

throw new Error(format(ERROR.MALFORMED\_UTF, [byte1.toString(16), byte2.toString(16), byte3.toString(16), byte4.toString(16)]));

}

}

}

if (utf16 > 0xFFFF) // 4 byte character - express as a surrogate pair

{

utf16 -= 0x10000;

output += String.fromCharCode(0xD800 + (utf16 >> 10)); // lead character

utf16 = 0xDC00 + (utf16 & 0x3FF); // trail character

}

output += String.fromCharCode(utf16);

}

return output;

}

/\*\* @ignore Repeat keepalive requests, monitor responses.\*/

var Pinger = function (client, window, keepAliveInterval) {

this.\_client = client;

this.\_window = window;

this.\_keepAliveInterval = keepAliveInterval \* 1000;

this.isReset = false;

var pingReq = new WireMessage(MESSAGE\_TYPE.PINGREQ).encode();

var doTimeout = function (pinger) {

return function () {

return doPing.apply(pinger);

};

};

/\*\* @ignore \*/

var doPing = function () {

if (!this.isReset) {

this.\_client.\_trace("Pinger.doPing", "Timed out");

this.\_client.\_disconnected(ERROR.PING\_TIMEOUT.code, format(ERROR.PING\_TIMEOUT));

} else {

this.isReset = false;

this.\_client.\_trace("Pinger.doPing", "send PINGREQ");

this.\_client.socket.send(pingReq);

this.timeout = this.\_window.setTimeout(doTimeout(this), this.\_keepAliveInterval);

}

}

this.reset = function () {

this.isReset = true;

this.\_window.clearTimeout(this.timeout);

if (this.\_keepAliveInterval > 0)

this.timeout = setTimeout(doTimeout(this), this.\_keepAliveInterval);

}

this.cancel = function () {

this.\_window.clearTimeout(this.timeout);

}

};

/\*\* @ignore Monitor request completion. \*/

var Timeout = function (client, window, timeoutSeconds, action, args) {

this.\_window = window;

if (!timeoutSeconds)

timeoutSeconds = 30;

var doTimeout = function (action, client, args) {

return function () {

return action.apply(client, args);

};

};

this.timeout = setTimeout(doTimeout(action, client, args), timeoutSeconds \* 1000);

this.cancel = function () {

this.\_window.clearTimeout(this.timeout);

}

};

/\*

\* Internal implementation of the Websockets MQTT V3.1 client.

\*

\* @name Messaging.ClientImpl @constructor

\* @param {String} host the DNS nameof the webSocket host.

\* @param {Number} port the port number for that host.

\* @param {String} clientId the MQ client identifier.

\*/

var ClientImpl = function (host, port, clientId) {

// Check dependencies are satisfied in this browser.

if (!("WebSocket" in global && global["WebSocket"] !== null)) {

throw new Error(format(ERROR.UNSUPPORTED, ["WebSocket"]));

}

if (!("localStorage" in global && global["localStorage"] !== null)) {

throw new Error(format(ERROR.UNSUPPORTED, ["localStorage"]));

}

if (!("ArrayBuffer" in global && global["ArrayBuffer"] !== null)) {

throw new Error(format(ERROR.UNSUPPORTED, ["ArrayBuffer"]));

}

this.\_trace("Messaging.Client", host, port, clientId);

this.host = host;

this.port = port;

this.clientId = clientId;

// Local storagekeys are qualified with the following string.

this.\_localKey = host + ":" + port + ":" + clientId + ":";

// Create private instance-only message queue

// Internal queue of messages to be sent, in sending order.

this.\_msg\_queue = [];

// Messages we have sent and are expecting a response for, indexed by their respective message ids.

this.\_sentMessages = {};

// Messages we have received and acknowleged and are expecting a confirm message for

// indexed by their respective message ids.

this.\_receivedMessages = {};

// Internal list of callbacks to be executed when messages

// have been successfully sent over web socket, e.g. disconnect

// when it doesn't have to wait for ACK, just message is dispatched.

this.\_notify\_msg\_sent = {};

// Unique identifier for SEND messages, incrementing

// counter as messages are sent.

this.\_message\_identifier = 1;

// Used to determine the transmission sequence of stored sent messages.

this.\_sequence = 0;

// Load the local state, if any, from the saved version, only restore state relevant to this client.

for (key in localStorage)

if (key.indexOf("Sent:" + this.\_localKey) == 0

|| key.indexOf("Received:" + this.\_localKey) == 0)

this.restore(key);

};

// Messaging Client public instance members.

ClientImpl.prototype.host;

ClientImpl.prototype.port;

ClientImpl.prototype.clientId;

// Messaging Client private instance members.

ClientImpl.prototype.socket;

/\* true once we have received an acknowledgement to a CONNECT packet. \*/

ClientImpl.prototype.connected = false;

/\* The largest message identifier allowed, may not be larger than 2\*\*16 but

\* if set smaller reduces the maximum number of outbound messages allowed.

\*/

ClientImpl.prototype.maxMessageIdentifier = 65536;

ClientImpl.prototype.connectOptions;

ClientImpl.prototype.hostIndex;

ClientImpl.prototype.onConnectionLost;

ClientImpl.prototype.onMessageDelivered;

ClientImpl.prototype.onMessageArrived;

ClientImpl.prototype.\_msg\_queue = null;

ClientImpl.prototype.\_connectTimeout;

/\* The sendPinger monitors how long we allow before we send data to prove to the server that we are alive. \*/

ClientImpl.prototype.sendPinger = null;

/\* The receivePinger monitors how long we allow before we require evidence that the server is alive. \*/

ClientImpl.prototype.receivePinger = null;

ClientImpl.prototype.\_traceBuffer = null;

ClientImpl.prototype.\_MAX\_TRACE\_ENTRIES = 100;

ClientImpl.prototype.connect = function (connectOptions) {

var connectOptionsMasked = this.\_traceMask(connectOptions, "password");

this.\_trace("Client.connect", connectOptionsMasked, this.socket, this.connected);

if (this.connected)

throw new Error(format(ERROR.INVALID\_STATE, ["already connected"]));

if (this.socket)

throw new Error(format(ERROR.INVALID\_STATE, ["already connected"]));

this.connectOptions = connectOptions;

if (connectOptions.hosts) {

this.hostIndex = 0;

this.\_doConnect(connectOptions.hosts[0], connectOptions.ports[0]);

} else {

this.\_doConnect(this.host, this.port);

}

};

ClientImpl.prototype.subscribe = function (filter, subscribeOptions) {

this.\_trace("Client.subscribe", filter, subscribeOptions);

if (!this.connected)

throw new Error(format(ERROR.INVALID\_STATE, ["not connected"]));

var wireMessage = new WireMessage(MESSAGE\_TYPE.SUBSCRIBE);

wireMessage.topics = [filter];

if (subscribeOptions.qos != undefined)

wireMessage.requestedQos = [subscribeOptions.qos];

else

wireMessage.requestedQos = [0];

if (subscribeOptions.onSuccess) {

wireMessage.callback = function () {

subscribeOptions.onSuccess({invocationContext: subscribeOptions.invocationContext});

};

}

if (subscribeOptions.timeout) {

wireMessage.timeOut = new Timeout(this, window, subscribeOptions.timeout, subscribeOptions.onFailure

, [

{invocationContext: subscribeOptions.invocationContext,

errorCode: ERROR.SUBSCRIBE\_TIMEOUT.code,

errorMessage: format(ERROR.SUBSCRIBE\_TIMEOUT)}

]);

}

// All subscriptions return a SUBACK.

this.\_requires\_ack(wireMessage);

this.\_schedule\_message(wireMessage);

};

/\*\* @ignore \*/

ClientImpl.prototype.unsubscribe = function (filter, unsubscribeOptions) {

this.\_trace("Client.unsubscribe", filter, unsubscribeOptions);

if (!this.connected)

throw new Error(format(ERROR.INVALID\_STATE, ["not connected"]));

var wireMessage = new WireMessage(MESSAGE\_TYPE.UNSUBSCRIBE);

wireMessage.topics = [filter];

if (unsubscribeOptions.onSuccess) {

wireMessage.callback = function () {

unsubscribeOptions.onSuccess({invocationContext: unsubscribeOptions.invocationContext});

};

}

if (unsubscribeOptions.timeout) {

wireMessage.timeOut = new Timeout(this, window, unsubscribeOptions.timeout, unsubscribeOptions.onFailure

, [

{invocationContext: unsubscribeOptions.invocationContext,

errorCode: ERROR.UNSUBSCRIBE\_TIMEOUT.code,

errorMessage: format(ERROR.UNSUBSCRIBE\_TIMEOUT)}

]);

}

// All unsubscribes return a SUBACK.

this.\_requires\_ack(wireMessage);

this.\_schedule\_message(wireMessage);

};

ClientImpl.prototype.send = function (message) {

this.\_trace("Client.send", message);

if (!this.connected)

throw new Error(format(ERROR.INVALID\_STATE, ["not connected"]));

wireMessage = new WireMessage(MESSAGE\_TYPE.PUBLISH);

wireMessage.payloadMessage = message;

if (message.qos > 0)

this.\_requires\_ack(wireMessage);

else if (this.onMessageDelivered)

this.\_notify\_msg\_sent[wireMessage] = this.onMessageDelivered(wireMessage.payloadMessage);

this.\_schedule\_message(wireMessage);

};

ClientImpl.prototype.disconnect = function () {

this.\_trace("Client.disconnect");

if (!this.socket)

throw new Error(format(ERROR.INVALID\_STATE, ["not connecting or connected"]));

wireMessage = new WireMessage(MESSAGE\_TYPE.DISCONNECT);

// Run the disconnected call back as soon as the message has been sent,

// in case of a failure later on in the disconnect processing.

// as a consequence, the \_disconected call back may be run several times.

this.\_notify\_msg\_sent[wireMessage] = scope(this.\_disconnected, this);

this.\_schedule\_message(wireMessage);

};

ClientImpl.prototype.getTraceLog = function () {

if (this.\_traceBuffer !== null) {

this.\_trace("Client.getTraceLog", new Date());

this.\_trace("Client.getTraceLog in flight messages", this.\_sentMessages.length);

for (key in this.\_sentMessages)

this.\_trace("\_sentMessages ", key, this.\_sentMessages[key]);

for (key in this.\_receivedMessages)

this.\_trace("\_receivedMessages ", key, this.\_receivedMessages[key]);

return this.\_traceBuffer;

}

};

ClientImpl.prototype.startTrace = function () {

if (this.\_traceBuffer === null) {

this.\_traceBuffer = [];

}

this.\_trace("Client.startTrace", new Date(), version);

};

ClientImpl.prototype.stopTrace = function () {

delete this.\_traceBuffer;

};

ClientImpl.prototype.\_doConnect = function (host, port) {

// When the socket is open, this client will send the CONNECT WireMessage using the saved parameters.

if (this.connectOptions.useSSL)

wsurl = ["wss://", host, ":", port, "/mqtt"].join("");

else

wsurl = ["ws://", host, ":", port, "/mqtt"].join("");

this.connected = false;

this.socket = new WebSocket(wsurl, 'mqttv3.1');

this.socket.binaryType = 'arraybuffer';

this.socket.onopen = scope(this.\_on\_socket\_open, this);

this.socket.onmessage = scope(this.\_on\_socket\_message, this);

this.socket.onerror = scope(this.\_on\_socket\_error, this);

this.socket.onclose = scope(this.\_on\_socket\_close, this);

this.sendPinger = new Pinger(this, window, this.connectOptions.keepAliveInterval);

this.receivePinger = new Pinger(this, window, this.connectOptions.keepAliveInterval);

this.\_connectTimeout = new Timeout(this, window, this.connectOptions.timeout, this.\_disconnected, [ERROR.CONNECT\_TIMEOUT.code, format(ERROR.CONNECT\_TIMEOUT)]);

};

// Schedule a new message to be sent over the WebSockets

// connection. CONNECT messages cause WebSocket connection

// to be started. All other messages are queued internally

// until this has happened. When WS connection starts, process

// all outstanding messages.

ClientImpl.prototype.\_schedule\_message = function (message) {

this.\_msg\_queue.push(message);

// Process outstanding messages in the queue if we have an open socket, and have received CONNACK.

if (this.connected) {

this.\_process\_queue();

}

};

ClientImpl.prototype.store = function (prefix, wireMessage) {

storedMessage = {type: wireMessage.type, messageIdentifier: wireMessage.messageIdentifier, version: 1};

switch (wireMessage.type) {

case MESSAGE\_TYPE.PUBLISH:

if (wireMessage.pubRecReceived)

storedMessage.pubRecReceived = true;

// Convert the payload to a hex string.

storedMessage.payloadMessage = {};

var hex = "";

var messageBytes = wireMessage.payloadMessage.payloadBytes;

for (var i = 0; i < messageBytes.length; i++) {

if (messageBytes[i] <= 0xF)

hex = hex + "0" + messageBytes[i].toString(16);

else

hex = hex + messageBytes[i].toString(16);

}

storedMessage.payloadMessage.payloadHex = hex;

storedMessage.payloadMessage.qos = wireMessage.payloadMessage.qos;

storedMessage.payloadMessage.destinationName = wireMessage.payloadMessage.destinationName;

if (wireMessage.payloadMessage.duplicate)

storedMessage.payloadMessage.duplicate = true;

if (wireMessage.payloadMessage.retained)

storedMessage.payloadMessage.retained = true;

// Add a sequence number to sent messages.

if (prefix.indexOf("Sent:") == 0) {

if (wireMessage.sequence === undefined)

wireMessage.sequence = ++this.\_sequence;

storedMessage.sequence = wireMessage.sequence;

}

break;

default:

throw Error(format(ERROR.INVALID\_STORED\_DATA, [key, storedMessage]));

}

localStorage.setItem(prefix + this.\_localKey + wireMessage.messageIdentifier, JSON.stringify(storedMessage));

};

ClientImpl.prototype.restore = function (key) {

var value = localStorage.getItem(key);

var storedMessage = JSON.parse(value);

var wireMessage = new WireMessage(storedMessage.type, storedMessage);

switch (storedMessage.type) {

case MESSAGE\_TYPE.PUBLISH:

// Replace the payload message with a Message object.

var hex = storedMessage.payloadMessage.payloadHex;

var buffer = new ArrayBuffer((hex.length) / 2);

var byteStream = new Uint8Array(buffer);

var i = 0;

while (hex.length >= 2) {

var x = parseInt(hex.substring(0, 2), 16);

hex = hex.substring(2, hex.length);

byteStream[i++] = x;

}

var payloadMessage = new Messaging.Message(byteStream);

payloadMessage.qos = storedMessage.payloadMessage.qos;

payloadMessage.destinationName = storedMessage.payloadMessage.destinationName;

if (storedMessage.payloadMessage.duplicate)

payloadMessage.duplicate = true;

if (storedMessage.payloadMessage.retained)

payloadMessage.retained = true;

wireMessage.payloadMessage = payloadMessage;

break;

default:

throw Error(format(ERROR.INVALID\_STORED\_DATA, [key, value]));

}

if (key.indexOf("Sent:" + this.\_localKey) == 0) {

this.\_sentMessages[wireMessage.messageIdentifier] = wireMessage;

} else if (key.indexOf("Received:" + this.\_localKey) == 0) {

this.\_receivedMessages[wireMessage.messageIdentifier] = wireMessage;

}

};

ClientImpl.prototype.\_process\_queue = function () {

var message = null;

// Process messages in order they were added

var fifo = this.\_msg\_queue.reverse();

// Send all queued messages down socket connection

while ((message = fifo.pop())) {

this.\_socket\_send(message);

// Notify listeners that message was successfully sent

if (this.\_notify\_msg\_sent[message]) {

this.\_notify\_msg\_sent[message]();

delete this.\_notify\_msg\_sent[message];

}

}

};

/\*\*

\* @ignore

\* Expect an ACK response for this message. Add message to the set of in progress

\* messages and set an unused identifier in this message.

\*/

ClientImpl.prototype.\_requires\_ack = function (wireMessage) {

var messageCount = Object.keys(this.\_sentMessages).length;

if (messageCount > this.maxMessageIdentifier)

throw Error("Too many messages:" + messageCount);

while (this.\_sentMessages[this.\_message\_identifier] !== undefined) {

this.\_message\_identifier++;

}

wireMessage.messageIdentifier = this.\_message\_identifier;

this.\_sentMessages[wireMessage.messageIdentifier] = wireMessage;

if (wireMessage.type === MESSAGE\_TYPE.PUBLISH) {

this.store("Sent:", wireMessage);

}

if (this.\_message\_identifier === this.maxMessagIdentifier) {

this.\_message\_identifier = 1;

}

};

/\*\*

\* @ignore

\* Called when the underlying websocket has been opened.

\*/

ClientImpl.prototype.\_on\_socket\_open = function () {

// Create the CONNECT message object.

var wireMessage = new WireMessage(MESSAGE\_TYPE.CONNECT, this.connectOptions);

wireMessage.clientId = this.clientId;

this.\_socket\_send(wireMessage);

};

/\*\*

\* @ignore

\* Called when the underlying websocket has received a complete packet.

\*/

ClientImpl.prototype.\_on\_socket\_message = function (event) {

this.\_trace("Client.\_on\_socket\_message", event.data);

// Reset the receive ping timer, we now have evidence the server is alive.

this.receivePinger.reset();

var byteArray = new Uint8Array(event.data);

try {

var wireMessage = decodeMessage(byteArray);

} catch (error) {

this.\_disconnected(ERROR.INTERNAL\_ERROR.code, format(ERROR.INTERNAL\_ERROR, [error.message]));

return;

}

this.\_trace("Client.\_on\_socket\_message", wireMessage);

switch (wireMessage.type) {

case MESSAGE\_TYPE.CONNACK:

this.\_connectTimeout.cancel();

// If we have started using clean session then clear up the local state.

if (this.connectOptions.cleanSession) {

for (key in this.\_sentMessages) {

var sentMessage = this.\_sentMessages[key];

localStorage.removeItem("Sent:" + this.\_localKey + sentMessage.messageIdentifier);

}

this.\_sentMessages = {};

for (key in this.\_receivedMessages) {

var receivedMessage = this.\_receivedMessages[key];

localStorage.removeItem("Received:" + this.\_localKey + receivedMessage.messageIdentifier);

}

this.\_receivedMessages = {};

}

// Client connected and ready for business.

if (wireMessage.returnCode === 0) {

this.connected = true;

// Jump to the end of the list of hosts and stop looking for a good host.

if (this.connectOptions.hosts)

this.hostIndex = this.connectOptions.hosts.length;

} else {

this.\_disconnected(ERROR.CONNACK\_RETURNCODE.code, format(ERROR.CONNACK\_RETURNCODE, [wireMessage.returnCode, CONNACK\_RC[wireMessage.returnCode]]));

break;

}

// Resend messages.

var sequencedMessages = new Array();

for (var msgId in this.\_sentMessages) {

if (this.\_sentMessages.hasOwnProperty(msgId))

sequencedMessages.push(this.\_sentMessages[msgId]);

}

// Sort sentMessages into the original sent order.

var sequencedMessages = sequencedMessages.sort(function (a, b) {

return a.sequence - b.sequence;

});

for (var i = 0, len = sequencedMessages.length; i < len; i++) {

var sentMessage = sequencedMessages[i];

if (sentMessage.type == MESSAGE\_TYPE.PUBLISH && sentMessage.pubRecReceived) {

var pubRelMessage = new WireMessage(MESSAGE\_TYPE.PUBREL, {messageIdentifier: sentMessage.messageIdentifier});

this.\_schedule\_message(pubRelMessage);

} else {

this.\_schedule\_message(sentMessage);

}

;

}

// Execute the connectOptions.onSuccess callback if there is one.

if (this.connectOptions.onSuccess) {

this.connectOptions.onSuccess({invocationContext: this.connectOptions.invocationContext});

}

// Process all queued messages now that the connection is established.

this.\_process\_queue();

break;

case MESSAGE\_TYPE.PUBLISH:

this.\_receivePublish(wireMessage);

break;

case MESSAGE\_TYPE.PUBACK:

var sentMessage = this.\_sentMessages[wireMessage.messageIdentifier];

// If this is a re flow of a PUBACK after we have restarted receivedMessage will not exist.

if (sentMessage) {

delete this.\_sentMessages[wireMessage.messageIdentifier];

localStorage.removeItem("Sent:" + this.\_localKey + wireMessage.messageIdentifier);

if (this.onMessageDelivered)

this.onMessageDelivered(sentMessage.payloadMessage);

}

break;

case MESSAGE\_TYPE.PUBREC:

var sentMessage = this.\_sentMessages[wireMessage.messageIdentifier];

// If this is a re flow of a PUBREC after we have restarted receivedMessage will not exist.

if (sentMessage) {

sentMessage.pubRecReceived = true;

var pubRelMessage = new WireMessage(MESSAGE\_TYPE.PUBREL, {messageIdentifier: wireMessage.messageIdentifier});

this.store("Sent:", sentMessage);

this.\_schedule\_message(pubRelMessage);

}

break;

case MESSAGE\_TYPE.PUBREL:

var receivedMessage = this.\_receivedMessages[wireMessage.messageIdentifier];

localStorage.removeItem("Received:" + this.\_localKey + wireMessage.messageIdentifier);

// If this is a re flow of a PUBREL after we have restarted receivedMessage will not exist.

if (receivedMessage) {

this.\_receiveMessage(receivedMessage);

delete this.\_receivedMessages[wireMessage.messageIdentifier];

}

// Always flow PubComp, we may have previously flowed PubComp but the server lost it and restarted.

pubCompMessage = new WireMessage(MESSAGE\_TYPE.PUBCOMP, {messageIdentifier: wireMessage.messageIdentifier});

this.\_schedule\_message(pubCompMessage);

break;

case MESSAGE\_TYPE.PUBCOMP:

var sentMessage = this.\_sentMessages[wireMessage.messageIdentifier];

delete this.\_sentMessages[wireMessage.messageIdentifier];

localStorage.removeItem("Sent:" + this.\_localKey + wireMessage.messageIdentifier);

if (this.onMessageDelivered)

this.onMessageDelivered(sentMessage.payloadMessage);

break;

case MESSAGE\_TYPE.SUBACK:

var sentMessage = this.\_sentMessages[wireMessage.messageIdentifier];

if (sentMessage) {

if (sentMessage.timeOut)

sentMessage.timeOut.cancel();

if (sentMessage.callback) {

sentMessage.callback();

}

delete this.\_sentMessages[wireMessage.messageIdentifier];

}

break;

case MESSAGE\_TYPE.UNSUBACK:

var sentMessage = this.\_sentMessages[wireMessage.messageIdentifier];

if (sentMessage) {

if (sentMessage.timeOut)

sentMessage.timeOut.cancel();

if (sentMessage.callback) {

sentMessage.callback();

}

delete this.\_sentMessages[wireMessage.messageIdentifier];

}

break;

case MESSAGE\_TYPE.PINGRESP:

/\* The sendPinger or receivePinger may have sent a ping, the receivePinger has already been reset. \*/

this.sendPinger.reset();

break;

case MESSAGE\_TYPE.DISCONNECT:

// Clients do not expect to receive disconnect packets.

this.\_disconnected(ERROR.INVALID\_MQTT\_MESSAGE\_TYPE.code, format(ERROR.INVALID\_MQTT\_MESSAGE\_TYPE, [wireMessage.type]));

break;

default:

this.\_disconnected(ERROR.INVALID\_MQTT\_MESSAGE\_TYPE.code, format(ERROR.INVALID\_MQTT\_MESSAGE\_TYPE, [wireMessage.type]));

}

;

};

/\*\* @ignore \*/

ClientImpl.prototype.\_on\_socket\_error = function (error) {

this.\_disconnected(ERROR.SOCKET\_ERROR.code, format(ERROR.SOCKET\_ERROR, [error.data]));

};

/\*\* @ignore \*/

ClientImpl.prototype.\_on\_socket\_close = function () {

this.\_disconnected(ERROR.SOCKET\_CLOSE.code, format(ERROR.SOCKET\_CLOSE));

};

/\*\* @ignore \*/

ClientImpl.prototype.\_socket\_send = function (wireMessage) {

if (wireMessage.type == 1) {

var wireMessageMasked = this.\_traceMask(wireMessage, "password");

this.\_trace("Client.\_socket\_send", wireMessageMasked);

}

else this.\_trace("Client.\_socket\_send", wireMessage);

this.socket.send(wireMessage.encode());

/\* We have proved to the server we are alive. \*/

this.sendPinger.reset();

};

/\*\* @ignore \*/

ClientImpl.prototype.\_receivePublish = function (wireMessage) {

switch (wireMessage.payloadMessage.qos) {

case "undefined":

case 0:

this.\_receiveMessage(wireMessage);

break;

case 1:

var pubAckMessage = new WireMessage(MESSAGE\_TYPE.PUBACK, {messageIdentifier: wireMessage.messageIdentifier});

this.\_schedule\_message(pubAckMessage);

this.\_receiveMessage(wireMessage);

break;

case 2:

this.\_receivedMessages[wireMessage.messageIdentifier] = wireMessage;

this.store("Received:", wireMessage);

var pubRecMessage = new WireMessage(MESSAGE\_TYPE.PUBREC, {messageIdentifier: wireMessage.messageIdentifier});

this.\_schedule\_message(pubRecMessage);

break;

default:

throw Error("Invaild qos=" + wireMmessage.payloadMessage.qos);

}

;

};

/\*\* @ignore \*/

ClientImpl.prototype.\_receiveMessage = function (wireMessage) {

if (this.onMessageArrived) {

this.onMessageArrived(wireMessage.payloadMessage);

}

};

/\*\*

\* @ignore

\* Client has disconnected either at its own request or because the server

\* or network disconnected it. Remove all non-durable state.

\* @param {errorCode} [number] the error number.

\* @param {errorText} [string] the error text.

\*/

ClientImpl.prototype.\_disconnected = function (errorCode, errorText) {

this.\_trace("Client.\_disconnected", errorCode, errorText);

this.sendPinger.cancel();

this.receivePinger.cancel();

if (this.\_connectTimeout)

this.\_connectTimeout.cancel();

// Clear message buffers.

this.\_msg\_queue = [];

this.\_notify\_msg\_sent = {};

if (this.socket) {

// Cancel all socket callbacks so that they cannot be driven again by this socket.

this.socket.onopen = null;

this.socket.onmessage = null;

this.socket.onerror = null;

this.socket.onclose = null;

if (this.socket.readyState === 1)

this.socket.close();

delete this.socket;

}

if (this.connectOptions.hosts && this.hostIndex < this.connectOptions.hosts.length - 1) {

// Try the next host.

this.hostIndex++;

this.\_doConnect(this.connectOptions.hosts[this.hostIndex], this.connectOptions.ports[this.hostIndex]);

} else {

if (errorCode === undefined) {

errorCode = ERROR.OK.code;

errorText = format(ERROR.OK);

}

// Run any application callbacks last as they may attempt to reconnect and hence create a new socket.

if (this.connected) {

this.connected = false;

// Execute the connectionLostCallback if there is one, and we were connected.

if (this.onConnectionLost)

this.onConnectionLost({errorCode: errorCode, errorMessage: errorText});

} else {

// Otherwise we never had a connection, so indicate that the connect has failed.

if (this.connectOptions.onFailure)

this.connectOptions.onFailure({invocationContext: this.connectOptions.invocationContext, errorCode: errorCode, errorMessage: errorText});

}

}

};

/\*\* @ignore \*/

ClientImpl.prototype.\_trace = function () {

if (this.\_traceBuffer !== null) {

for (var i = 0, max = arguments.length; i < max; i++) {

if (this.\_traceBuffer.length == this.\_MAX\_TRACE\_ENTRIES) {

this.\_traceBuffer.shift();

}

if (i === 0) this.\_traceBuffer.push(arguments[i]);

else if (typeof arguments[i] === "undefined") this.\_traceBuffer.push(arguments[i]);

else this.\_traceBuffer.push(" " + JSON.stringify(arguments[i]));

}

;

}

;

};

/\*\* @ignore \*/

ClientImpl.prototype.\_traceMask = function (traceObject, masked) {

var traceObjectMasked = {};

for (var attr in traceObject) {

if (traceObject.hasOwnProperty(attr)) {

if (attr == masked)

traceObjectMasked[attr] = "\*\*\*\*\*\*";

else

traceObjectMasked[attr] = traceObject[attr];

}

}

return traceObjectMasked;

};

// ------------------------------------------------------------------------

// Public Programming interface.

// ------------------------------------------------------------------------

/\*\*

\* The JavaScript application communicates to the server using a Messaging.Client object.

\* <p>

\* Other programming languages,

\* <a href="/clients/java/doc/javadoc/com/ibm/micro/client/mqttv3/MqttClient.html"><big>Java</big></a>,

\* <a href="/clients/c/doc/html/index.html"><big>C</big></a>.

\* <p>

\* Most applications will create just one Client object and then call its connect() method,

\* however applications can create more than one Client object if they wish.

\* In this case the combination of host, port and clientId attributes must be different for each Client object.

\* <p>

\* The send, subscribe and unsubscribe methods are implemented as asynchronous JavaScript methods

\* (even though the underlying protocol exchange might be synchronous in nature).

\* This means they signal their completion by calling back to the application,

\* via Success or Failure callback functions provided by the application on the method in question.

\* Such callbacks are called at most once per method invocation and do not persist beyond the lifetime

\* of the script that made the invocation.

\* <p>

\* In contrast there are some callback functions <i> most notably onMessageArrived</i>

\* that are defined on the Messaging.Client object.

\* These may get called multiple times, and aren't directly related to specific method invocations made by the client.

\*

\* @name Messaging.Client

\*

\* @constructor

\* Creates a Messaging.Client object that can be used to communicate with a Messaging server.

\*

\* @param {string} host the address of the messaging server, as a DNS name or dotted decimal IP address.

\* @param {number} port the port number in the host to connect to.

\* @param {string} clientId the Messaging client identifier, between 1 and 23 characters in length.

\*

\* @property {string} host <i>read only</i> the server's DNS hostname or dotted decimal IP address.

\* @property {number} port <i>read only</i> the server's port.

\* @property {string} clientId <i>read only</i> used when connecting to the server.

\* @property {function} onConnectionLost called when a connection has been lost,

\* after a connect() method has succeeded.

\* Establish the call back used when a connection has been lost. The connection may be

\* lost because the client initiates a disconnect or because the server or network

\* cause the client to be disconnected. The disconnect call back may be called without

\* the connectionComplete call back being invoked if, for example the client fails to

\* connect.

\* A single response object parameter is passed to the onConnectionLost callback containing the following fields:

\* <ol>

\* <li>errorCode

\* <li>errorMessage

\* </ol>

\* @property {function} onMessageDelivered called when a message has been delivered.

\* All processing that this Client will ever do has been completed. So, for example,

\* in the case of a Qos=2 message sent by this client, the PubComp flow has been received from the server

\* and the message has been removed from persistent storage before this callback is invoked.

\* Parameters passed to the onMessageDelivered callback are:

\* <ol>

\* <li>Messaging.Message that was delivered.

\* </ol>

\* @property {function} onMessageArrived called when a message has arrived in this Messaging.client.

\* Parameters passed to the onMessageArrived callback are:

\* <ol>

\* <li>Messaging.Message that has arrived.

\* </ol>

\*/

var Client = function (host, port, clientId) {

if (typeof host !== "string")

throw new Error(format(ERROR.INVALID\_TYPE, [typeof host, "host"]));

if (typeof port !== "number" || port < 0)

throw new Error(format(ERROR.INVALID\_TYPE, [typeof port, "port"]));

var clientIdLength = 0;

for (var i = 0; i < clientId.length; i++) {

var charCode = clientId.charCodeAt(i);

if (0xD800 <= charCode && charCode <= 0xDBFF) {

i++; // Surrogate pair.

}

clientIdLength++;

}

if (typeof clientId !== "string" || clientIdLength < 1 | clientIdLength > 23)

throw new Error(format(ERROR.INVALID\_ARGUMENT, [clientId, "clientId"]));

var client = new ClientImpl(host, port, clientId);

this.\_getHost = function () {

return client.host;

};

this.\_setHost = function () {

throw new Error(format(ERROR.UNSUPPORTED\_OPERATION));

};

this.\_getPort = function () {

return client.port;

};

this.\_setPort = function () {

throw new Error(format(ERROR.UNSUPPORTED\_OPERATION));

};

this.\_getClientId = function () {

return client.clientId;

};

this.\_setClientId = function () {

throw new Error(format(ERROR.UNSUPPORTED\_OPERATION));

};

this.\_getOnConnectionLost = function () {

return client.onConnectionLost;

};

this.\_setOnConnectionLost = function (newOnConnectionLost) {

if (typeof newOnConnectionLost === "function")

client.onConnectionLost = newOnConnectionLost;

else

throw new Error(format(ERROR.INVALID\_TYPE, [typeof newOnConnectionLost, "onConnectionLost"]));

};

this.\_getOnMessageDelivered = function () {

return client.onMessageDelivered;

};

this.\_setOnMessageDelivered = function (newOnMessageDelivered) {

if (typeof newOnMessageDelivered === "function")

client.onMessageDelivered = newOnMessageDelivered;

else

throw new Error(format(ERROR.INVALID\_TYPE, [typeof newOnMessageDelivered, "onMessageDelivered"]));

};

this.\_getOnMessageArrived = function () {

return client.onMessageArrived;

};

this.\_setOnMessageArrived = function (newOnMessageArrived) {

if (typeof newOnMessageArrived === "function")

client.onMessageArrived = newOnMessageArrived;

else

throw new Error(format(ERROR.INVALID\_TYPE, [typeof newOnMessageArrived, "onMessageArrived"]));

};

/\*\*

\* Connect this Messaging client to its server.

\*

\* @name Messaging.Client#connect

\* @function

\* @param {Object} [connectOptions] attributes used with the connection.

\* <p>

\* Properties of the connect options are:

\* @config {number} [timeout] If the connect has not succeeded within this number of seconds, it is deemed to have failed.

\* The default is 30 seconds.

\* @config {string} [userName] Authentication username for this connection.

\* @config {string} [password] Authentication password for this connection.

\* @config {Messaging.Message} [willMessage] sent by the server when the client disconnects abnormally.

\* @config {Number} [keepAliveInterval] the server disconnects this client if there is no activity for this

\* number of seconds. The default value of 60 seconds is assumed if not set.

\* @config {boolean} [cleanSession] if true(default) the client and server persistent state is deleted on successful connect.

\* @config {boolean} [useSSL] if present and true, use an SSL Websocket connection.

\* @config {object} [invocationContext] passed to the onSuccess callback or onFailure callback.

\* @config {function} [onSuccess] called when the connect acknowledgement has been received from the server.

\* A single response object parameter is passed to the onSuccess callback containing the following fields:

\* <ol>

\* <li>invocationContext as passed in to the onSuccess method in the connectOptions.

\* </ol>

\* @config {function} [onFailure] called when the connect request has failed or timed out.

\* A single response object parameter is passed to the onFailure callback containing the following fields:

\* <ol>

\* <li>invocationContext as passed in to the onFailure method in the connectOptions.

\* <li>errorCode a number indicating the nature of the error.

\* <li>errorMessage text describing the error.

\* </ol>

\* @config {Array} [hosts] If present this set of hostnames is tried in order in place

\* of the host and port paramater on the construtor. The hosts and the matching ports are tried one at at time in order until

\* one of then succeeds.

\* @config {Array} [ports] If present this set of ports matching the hosts.

\* @throws {InvalidState} if the client is not in disconnected state. The client must have received connectionLost

\* or disconnected before calling connect for a second or subsequent time.

\*/

this.connect = function (connectOptions) {

connectOptions = connectOptions || {};

validate(connectOptions, {timeout: "number",

userName: "string",

password: "string",

willMessage: "object",

keepAliveInterval: "number",

cleanSession: "boolean",

useSSL: "boolean",

invocationContext: "object",

onSuccess: "function",

onFailure: "function",

hosts: "object",

ports: "object"});

// If no keep alive interval is set, assume 60 seconds.

if (connectOptions.keepAliveInterval === undefined)

connectOptions.keepAliveInterval = 60;

if (connectOptions.willMessage) {

if (!(connectOptions.willMessage instanceof Message))

throw new Error(format(ERROR.INVALID\_TYPE, [connectOptions.willMessage, "connectOptions.willMessage"]));

// The will message must have a payload that can be represented as a string.

// Cause the willMessage to throw an exception if this is not the case.

connectOptions.willMessage.stringPayload;

if (typeof connectOptions.willMessage.destinationName === "undefined")

throw new Error(format(ERROR.INVALID\_TYPE, [typeof connectOptions.willMessage.destinationName, "connectOptions.willMessage.destinationName"]));

}

if (typeof connectOptions.cleanSession === "undefined")

connectOptions.cleanSession = true;

if (connectOptions.hosts) {

if (!connectOptions.ports)

throw new Error(format(ERROR.INVALID\_ARGUMENT, [connectOptions.ports, "connectOptions.ports"]));

if (!(connectOptions.hosts instanceof Array))

throw new Error(format(ERROR.INVALID\_ARGUMENT, [connectOptions.hosts, "connectOptions.hosts"]));

if (!(connectOptions.ports instanceof Array))

throw new Error(format(ERROR.INVALID\_ARGUMENT, [connectOptions.ports, "connectOptions.ports"]));

if (connectOptions.hosts.length < 1)

throw new Error(format(ERROR.INVALID\_ARGUMENT, [connectOptions.hosts, "connectOptions.hosts"]));

if (connectOptions.hosts.length != connectOptions.ports.length)

throw new Error(format(ERROR.INVALID\_ARGUMENT, [connectOptions.ports, "connectOptions.ports"]));

for (var i = 0; i < connectOptions.hosts.length; i++) {

if (typeof connectOptions.hosts[i] !== "string")

throw new Error(format(ERROR.INVALID\_TYPE, [typeof connectOptions.hosts[i], "connectOptions.hosts[" + i + "]"]));

if (typeof connectOptions.ports[i] !== "number" || connectOptions.ports[i] < 0)

throw new Error(format(ERROR.INVALID\_TYPE, [typeof connectOptions.ports[i], "connectOptions.ports[" + i + "]"]));

}

}

client.connect(connectOptions);

};

/\*\*

\* Subscribe for messages, request receipt of a copy of messages sent to the destinations described by the filter.

\*

\* @name Messaging.Client#subscribe

\* @function

\* @param {string} filter describing the destinations to receive messages from.

\* <br>

\* @param {object} [subscribeOptions] used to control the subscription, as follows:

\* <p>

\* @config {number} [qos] the maiximum qos of any publications sent as a result of making this subscription.

\* @config {object} [invocationContext] passed to the onSuccess callback or onFailure callback.

\* @config {function} [onSuccess] called when the subscribe acknowledgement has been received from the server.

\* A single response object parameter is passed to the onSuccess callback containing the following fields:

\* <ol>

\* <li>invocationContext if set in the subscribeOptions.

\* </ol>

\* @config {function} [onFailure] called when the subscribe request has failed or timed out.

\* A single response object parameter is passed to the onFailure callback containing the following fields:

\* <ol>

\* <li>invocationContext if set in the subscribeOptions.

\* <li>errorCode a number indicating the nature of the error.

\* <li>errorMessage text describing the error.

\* </ol>

\* @config {number} [timeout] which if present determines the number of seconds after which the onFailure calback is called

\* the presence of a timeout does not prevent the onSuccess callback from being called when the MQTT Suback is eventually received.

\* @throws {InvalidState} if the client is not in connected state.

\*/

this.subscribe = function (filter, subscribeOptions) {

if (typeof filter !== "string")

throw new Error("Invalid argument:" + filter);

subscribeOptions = subscribeOptions || {};

validate(subscribeOptions, {qos: "number",

invocationContext: "object",

onSuccess: "function",

onFailure: "function",

timeout: "number"

});

if (subscribeOptions.timeout && !subscribeOptions.onFailure)

throw new Error("subscribeOptions.timeout specified with no onFailure callback.");

if (typeof subscribeOptions.qos !== "undefined"

&& !(subscribeOptions.qos === 0 || subscribeOptions.qos === 1 || subscribeOptions.qos === 2 ))

throw new Error(format(ERROR.INVALID\_ARGUMENT, [subscribeOptions.qos, "subscribeOptions.qos"]));

client.subscribe(filter, subscribeOptions);

};

/\*\*

\* Unsubscribe for messages, stop receiving messages sent to destinations described by the filter.

\*

\* @name Messaging.Client#unsubscribe

\* @function

\* @param {string} filter describing the destinations to receive messages from.

\* @param {object} [unsubscribeOptions] used to control the subscription, as follows:

\* <p>

\* @config {object} [invocationContext] passed to the onSuccess callback or onFailure callback.

\* @config {function} [onSuccess] called when the unsubscribe acknowledgement has been receive dfrom the server.

\* A single response object parameter is passed to the onSuccess callback containing the following fields:

\* <ol>

\* <li>invocationContext if set in the unsubscribeOptions.

\* </ol>

\* @config {function} [onFailure] called when the unsubscribe request has failed or timed out.

\* A single response object parameter is passed to the onFailure callback containing the following fields:

\* <ol>

\* <li>invocationContext if set in the unsubscribeOptions.

\* <li>errorCode a number indicating the nature of the error.

\* <li>errorMessage text describing the error.

\* </ol>

\* @config {number} [timeout] which if present determines the number of seconds after which the onFailure callback is called, the

\* presence of a timeout does not prevent the onSuccess callback from being called when the MQTT UnSuback is eventually received.

\* @throws {InvalidState} if the client is not in connected state.

\*/

this.unsubscribe = function (filter, unsubscribeOptions) {

if (typeof filter !== "string")

throw new Error("Invalid argument:" + filter);

unsubscribeOptions = unsubscribeOptions || {};

validate(unsubscribeOptions, {invocationContext: "object",

onSuccess: "function",

onFailure: "function",

timeout: "number"

});

if (unsubscribeOptions.timeout && !unsubscribeOptions.onFailure)

throw new Error("unsubscribeOptions.timeout specified with no onFailure callback.");

client.unsubscribe(filter, unsubscribeOptions);

};

/\*\*

\* Send a message to the consumers of the destination in the Message.

\*

\* @name Messaging.Client#send

\* @function

\* @param {Messaging.Message} message to send.

\* @throws {InvalidState} if the client is not in connected state.

\*/

this.send = function (message) {

if (!(message instanceof Message))

throw new Error("Invalid argument:" + typeof message);

if (typeof message.destinationName === "undefined")

throw new Error("Invalid parameter Message.destinationName:" + message.destinationName);

client.send(message);

};

/\*\*

\* Normal disconnect of this Messaging client from its server.

\*

\* @name Messaging.Client#disconnect

\* @function

\* @throws {InvalidState} if the client is not in connected or connecting state.

\*/

this.disconnect = function () {

client.disconnect();

};

/\*\*

\* Get the contents of the trace log.

\*

\* @name Messaging.Client#getTraceLog

\* @function

\* @return {Object[]} tracebuffer containing the time ordered trace records.

\*/

this.getTraceLog = function () {

return client.getTraceLog();

}

/\*\*

\* Start tracing.

\*

\* @name Messaging.Client#startTrace

\* @function

\*/

this.startTrace = function () {

client.startTrace();

};

/\*\*

\* Stop tracing.

\*

\* @name Messaging.Client#stopTrace

\* @function

\*/

this.stopTrace = function () {

client.stopTrace();

};

};

Client.prototype = {

get host() {

return this.\_getHost();

},

set host(newHost) {

this.\_setHost(newHost);

},

get port() {

return this.\_getPort();

},

set port(newPort) {

this.\_setPort(newPort);

},

get clientId() {

return this.\_getClientId();

},

set clientId(newClientId) {

this.\_setClientId(newClientId);

},

get onConnectionLost() {

return this.\_getOnConnectionLost();

},

set onConnectionLost(newOnConnectionLost) {

this.\_setOnConnectionLost(newOnConnectionLost);

},

get onMessageDelivered() {

return this.\_getOnMessageDelivered();

},

set onMessageDelivered(newOnMessageDelivered) {

this.\_setOnMessageDelivered(newOnMessageDelivered);

},

get onMessageArrived() {

return this.\_getOnMessageArrived();

},

set onMessageArrived(newOnMessageArrived) {

this.\_setOnMessageArrived(newOnMessageArrived);

}

};

/\*\*

\* An application message, sent or received.

\* <p>

\* Other programming languages,

\* <a href="/clients/java/doc/javadoc/com/ibm/micro/client/mqttv3/MqttMessage.html"><big>Java</big></a>,

\* <a href="/clients/c/doc/html/struct\_m\_q\_t\_t\_client\_\_message.html"><big>C</big></a>.

\* <p>

\* All attributes may be null, which implies the default values.

\*

\* @name Messaging.Message

\* @constructor

\* @param {String|ArrayBuffer} payload The message data to be sent.

\* <p>

\* @property {string} payloadString <i>read only</i> The payload as a string if the payload consists of valid UTF-8 characters.

\* @property {ArrayBuffer} payloadBytes <i>read only</i> The payload as an ArrayBuffer.

\* <p>

\* @property {string} destinationName <b>mandatory</b> The name of the destination to which the message is to be sent

\* (for messages about to be sent) or the name of the destination from which the message has been received.

\* (for messages received by the onMessage function).

\* <p>

\* @property {number} qos The Quality of Service used to deliver the message.

\* <dl>

\* <dt>0 Best effort (default).

\* <dt>1 At least once.

\* <dt>2 Exactly once.

\* </dl>

\* <p>

\* @property {Boolean} retained If true, the message is to be retained by the server and delivered

\* to both current and future subscriptions.

\* If false the server only delivers the message to current subscribers, this is the default for new Messages.

\* A received message has the retained boolean set to true if the message was published

\* with the retained boolean set to true

\* and the subscrption was made after the message has been published.

\* <p>

\* @property {Boolean} duplicate <i>read only</i> If true, this message might be a duplicate of one which has already been received.

\* This is only set on messages received from the server.

\*

\*/

var Message = function (newPayload) {

var payload;

if (typeof newPayload === "string"

|| newPayload instanceof ArrayBuffer

|| newPayload instanceof Int8Array

|| newPayload instanceof Uint8Array

|| newPayload instanceof Int16Array

|| newPayload instanceof Uint16Array

|| newPayload instanceof Int32Array

|| newPayload instanceof Uint32Array

|| newPayload instanceof Float32Array

|| newPayload instanceof Float64Array

) {

payload = newPayload;

} else {

throw (format(ERROR.INVALID\_ARGUMENT, [newPayload, "newPayload"]));

}

this.\_getPayloadString = function () {

if (typeof payload === "string")

return payload;

else

return parseUTF8(payload, 0, payload.length);

};

this.\_getPayloadBytes = function () {

if (typeof payload === "string") {

var buffer = new ArrayBuffer(UTF8Length(payload));

var byteStream = new Uint8Array(buffer);

stringToUTF8(payload, byteStream, 0);

return byteStream;

} else {

return payload;

}

;

};

var destinationName = undefined;

this.\_getDestinationName = function () {

return destinationName;

};

this.\_setDestinationName = function (newDestinationName) {

if (typeof newDestinationName === "string")

destinationName = newDestinationName;

else

throw new Error(format(ERROR.INVALID\_ARGUMENT, [newDestinationName, "newDestinationName"]));

};

var qos = 0;

this.\_getQos = function () {

return qos;

};

this.\_setQos = function (newQos) {

if (newQos === 0 || newQos === 1 || newQos === 2)

qos = newQos;

else

throw new Error("Invalid argument:" + newQos);

};

var retained = false;

this.\_getRetained = function () {

return retained;

};

this.\_setRetained = function (newRetained) {

if (typeof newRetained === "boolean")

retained = newRetained;

else

throw new Error(format(ERROR.INVALID\_ARGUMENT, [newRetained, "newRetained"]));

};

var duplicate = false;

this.\_getDuplicate = function () {

return duplicate;

};

this.\_setDuplicate = function (newDuplicate) {

duplicate = newDuplicate;

};

};

Message.prototype = {

get payloadString() {

return this.\_getPayloadString();

},

get payloadBytes() {

return this.\_getPayloadBytes();

},

get destinationName() {

return this.\_getDestinationName();

},

set destinationName(newDestinationName) {

this.\_setDestinationName(newDestinationName);

},

get qos() {

return this.\_getQos();

},

set qos(newQos) {

this.\_setQos(newQos);

},

get retained() {

return this.\_getRetained();

},

set retained(newRetained) {

this.\_setRetained(newRetained);

},

get duplicate() {

return this.\_getDuplicate();

},

set duplicate(newDuplicate) {

this.\_setDuplicate(newDuplicate);

}

};

// Module contents.

return {

Client: Client,

Message: Message

};

})(window);

# Приложение Г

**Руководство пользователя**

1. **Введение**
   1. **Область применения**

Областью применения программы является система умного дома для коммерческого или общего пользования. Основная цель программы – автоматизировать и облегчить использования разных устройств умного дома.

* 1. **Краткое содержание возможностей**

Данная система представляет собой контроллер на базе платформы Arduino с подключенными устройствами, управление которыми осуществляется с помощью мобильного приложения или личного кабинета на сайте «EMS INSYTE».

* 1. **Уровень подготовки пользователя**

Для управления системой необходима установка контроллера и устройств и подключение их к сети «Интернет», предоставляемая компанией.

* 1. **Перечень эксплуатационной документации**

Настоящее руководство пользователя и руководство программиста.

1. **Назначение и условия применения**
   1. **Виды деятельности, функции**

Программный продукт является контроллером для отправки и принятия изменений, подключенных устройств, от удаленного брокера компании «EMS INSYTE». Оно позволяет добавлять, модифицировать и удалять данные, выполнять запросы к данным. В целом программа служит инструментом для автоматизации устройств умного дома.

* 1. **Программные и аппаратные требования к системе**

Необходимо любое устройство с предустановленным браузером или приложением для отправки протоколов MQTT и выходом в интернет.

1. **Подготовка к работе**
   1. **Состав дистрибутива**

В состав дистрибутива входит контроллер Arduino и устройства умного дома.

* 1. **Запуск системы**

Запуск системы производит установщик, представленный от компании.

* 1. **Проверка работоспособности системы**

Программное обеспечение работоспособно, если в результате действий запуска горит красный светодиод.

1. **Описание операций**
   1. **Вход в систему**

Вход в систему производится автоматически при запуске системы с заранее выделенными логином и паролем.

* 1. **Внесение изменений в системе**

Условия выполнения: для добавления в базу данных информации о изменении переменных измените переменные с помощью мобильного приложения или на сайте «EMS INSYTE».

Подготовительные действия: авторизация в системе.

Основные действия: изменение переменных.

Заключительные действия не требуются.

Ресурсы, расходуемые на операцию: от секунды до минуты времени, ресурсы центрального процессора. Продолжительность выполнения операции зависит от интернет соединения и скорости отклика брокера.

* 1. **Просмотр информации об подключенных устройствах**

Условия выполнения: наличие подключения к сети «Интернет».

Подготовительные действия: авторизация в системе.

Основные действия: зайти в свой кабинет на сайте «EMS INSYTE», зайти в раздел устройства.

Заключительные действия: по завершении просмотра закрыть окно.

Ресурсы, расходуемые на операцию: от секунды до минуты времени, ресурсы центрального процессора. Продолжительность выполнения операции зависит от интернет соединения и скорости отклика брокера.

# Приложение Д

**Руководство разработчика**

1. **Общие сведения**
2. Документация: программная.
3. Предмет: программа для подключения к платформе ems insyte с помощью контроллера arduino.
4. Аудитория: С++ программисты, программисты контроллеров Arduino.
5. Задачи: обеспечить программисту возможность решать задачи, которые могут быть перед ним поставлены в отношении программы.
6. Стандарты: ГОСТ 19.504-79
7. **Цели и задачи**

Программный продукт по своему основному назначению является контроллером для отправки и принятия данных путем издатель-подписчик MQTT протокола.

1. **Назначение и условия применения программы**
   1. **Назначение программы**

Программный продукт создан для автоматизации и упрощения в управлении устройствами умного дома.

* 1. **Функции, выполняемые программой**

Программа работает с удаленным брокером по принципу издатель-подписчик MQTT протокола. Программный продукт позволяет изменять, добавлять или удалять данные устройств, подключенных к нему.

* 1. **Условия, необходимые для выполнения программы**

Необходимо любое устройство с предустановленным браузером или приложением для отправки протоколов MQTT и выходом в интернет.

1. **Характеристика программы**
   1. **Описание основных характеристик программы**

Скорость работы контроллера зависит от подключения к интернету и скорости ответа брокера.

* 1. **Особенности программы**

Программная система работает с удаленным брокером по сети «Интернет».

1. **Обращение к программе**

Запуск программы осуществляется сразу после установки и настройки контроллера.

1. **Входные и выходные данные**

Входные данные для контроллера от брокера и от контроллера к брокеру представлены в виде JSON списка.

1. **Сообщения**

При отсутствии какой-либо индикации проверить настройку контроллера.

При работе красного светодиода контроллер подключен к удаленному брокеру.

При работе желтого светодиода контроллер не подключен к удаленному брокеру либо отсутствует подключение к интернету.