

Электронная очередь, спецификация интерфейсов.

0. Электронная очередь (далее ЭО)

ЭО предоставляет два интерфейса: веб-сервис (API) и веб-приложение (UI).

Веб-сервис (API) работает в двух режимах: запрос-ответ (для ВС, описано в пункте 1) и постоянное соединение (для UI, доступное по адресу <http://server/qms/signalr> и предоставляющее события изменений в очереди `queueChanged` (reason, queueString, rotationTimeLimitParam)).

- 0.1. Внешние системы не уведомляются, ЭО обновляется.
- 0.2. Внешние системы не уведомляются, ЭО обновляется.
- 0.3. Внешние системы не уведомляются, ЭО обновляется.

1. Внешняя авторизованная система (далее ВС)

- 1.1. ВС посылает веб-сервису ЭО (конечной точке), веб-запрос о том, что Клиента необходимо поставить в очередь (сообщение **PUT** протокола REST), пример: <http://server/qms/api/queue/a10>

где параметр “[a10](#)” это номер парковочной карты, **уникально идентифицирующий Клиента в пределах очереди**.

Защита данного обмена осуществляется посредством само-подписанного сертификата X.509 по протоколу SSL. Защита будет подключена на завершающей стадии разработки, для чего разработчикам ВС направляется пошаговая [инструкция](#) по обмену открытыми ключами.

По адресу <http://server/qms/help> доступна веб-страница с описанием api:

Home API	
ParkingQueue.Service Web API Help Page	
Introduction	
ParkingQueue.Service использует REST-протокол. Ниже представлены доступные методы.	
Queue	
API	Description
GET api/Queue	Возвращает все элементы очереди
GET api/Queue?count={count}	Возвращает заданное количество элементов очереди
GET api/Queue/{id}	Возвращает информацию о заданном элементе в очереди
PUT api/Queue/{id}	Добавляет уникальный элемент в очередь
DELETE api/Queue/{id}?reason={reason}	Удаляет заданный элемент из очереди

- 1.2. ВС получает, в результате запроса 1.1, в заголовке код возврата “200 OK”, а в теле ответную строку-сообщение, в кодировке UTF-8, в формате JSON, формат:

<pre>{ "input": "2016-02-16T16:04:42", "parkingCard": "a10", "output": "2016-02-16T16:09:42", "rotation": 0, "position": 1 }</pre>	Время включения в очередь Номер парковочной карты Расчётное время до выдачи авто Количество ротаций Позиция в очереди
--	---

Данный формат ответа одинаков для всех запросов к сервису.

Возврат json-объекта “null” в теле ответа трактуется, как наличие заданной парковочной карты в очереди (один и тот же номер парковочной карты не может быть добавлен в очередь дважды).

Отсутствие ответа, либо иной ответ (без кода “200 OK”, либо без объекта с тем же идентификатором parkingCard в теле ответа (который отправлялся в запросе), трактуется, как исключительная ситуация (ошибка).

В этом случае необходимо отправить запрос к сервису повторно.

- 1.3. ВС запрашивает у ЭО разрешение выпустить Клиента (первый ли он в очереди?), аналогичный запросу 1.1 (сообщение **GET** протокола REST), пример:
<http://server/qms/api/queue/a10>
- 1.4. ВС получает ответ, аналогичный ответу 1.2
- 1.5. ВС посылает сообщение в ЭО об исключении Клиента из очереди, аналогичный запросу 1.1 (сообщение **DELETE** протокола REST), пример:
<http://server/qms/api/queue/a10?reason=vip>

Где параметр **reason** сообщает код причины исключения из очереди:

1.5.1. Пропуск Клиента согласно очереди = **peek**

1.5.2. Пропуск Клиента вне очереди = **vip**

1.5.3. Клиент не забрал машину и ВС вернула её на парковку = **time**

1.6. ВС получает ответную строку-сообщение, аналогичную ответу 1.2:

<pre>{ "input": "2016-02-16T16:04:42", "parkingCard": "a10", "output": "2016-02-16T16:09:42", "rotation": 0, "position": 1 "outputReasonId": "V" }</pre>	Время включения в очередь Номер парковочной карты Фактическое время выдачи авто Количество ротаций Позиция в очереди на момент выдачи авто Первая буква кода причины исключения из очереди (1.5.1 - 1.5.3)
--	---

Возврат json-объекта "null" в теле ответа трактуется, как отсутствие заданной парковочной карты в очереди (удалена ранее, либо вовсе не ставилась в очередь).

2. Клиент

2.1. На экран в зале выводится веб-приложение, страничка UI для Клиента, пример: <http://server/qms/ui>
строка запроса браузера и все его панели инструментов скрыты, обновление информации на экране происходит автоматически (UI Клиента использует websocket, UI Оператора и Администратора реализован, как asp.net mvc SPA).

Видимые Клиенту колонки в очереди:

1. Номер в очереди
2. Парковочная карта
3. Время ожидания
4. Количество опозданий

Макет экранной формы:

	Номер очереди	Парковочная карта	Время ожидания	Количество опозданий
00:08	1	ParkingCard_0147		
	2	ParkingCard_0148	13:33	
	3	ParkingCard_0149	13:38	
	4	ParkingCard_0150	13:43	
	5	ParkingCard_0146	13:48	1

Рекомендуемым устройством для вывода очереди является монитор (телевизор) диагональю от 60 дюймов, разрешением 1920 на 1080 пикселей и встроенными колонками, подключенный к компьютеру под управлением операционной системы windows 7 (или выше).

Также допустимо использование smart-tv телевизора (без использования компьютера) модельного ряда не позднее 2012 года со встроенным браузером, поддерживающим HTML 5. В этом случае необходимо протестировать способность встроенного браузера воспроизводить веб-сайты по произвольному адресу и скрывать строку подключения браузера и его панели инструментов.

2.2. Веб-приложение получает от сервера сообщение, обновляет страничку UI для Клиента и проигрывает звуковой [файл](#), при этом звук выводится на устройство воспроизведения, доступное браузеру на Клиентской стороне.

2.3. Аналогично 2.2

2.4. Аналогично 2.2

3. Оператор

3.1. Оператор запускает браузер и авторизуется в веб-приложении, пример: <http://server/qms/ui/operator> посредством windows-аутентификации (ввод логина/пароля не требуется, используется учётная запись домена)

3.2. Стартовая веб-страничка для Оператора аналогична страничке Клиента, но отображает очередь целиком, допускает фильтрацию по колонкам и содержит меню команд.

Видимые оператору колонки в очереди:

1. Номер парковочной карты (по клику - переход на детальную информацию 3.4)
2. Колонки из 2.1
3. Дата и время включения в ЭО

3.3. Оператор из 3.2 переходит на странички:

3.3.1. Со списком Клиентов, исключённых из ЭО, пример:

<http://server/qms/ui/operator/deQueueReport>

Колонки списка:

1. Номер парковочной карты (по клику - переход на детальную информацию 3.4)
2. Причина исключения из очереди (rotation, peek, vip, time)
3. Дата и время включения в ЭО

3.3.2. Со списком Клиентов и временем выхода из зоны ожидания, пример:

<http://server/qms/ui/operator/peekReport>

Колонки списка:

1. Номер парковочной карты (по клику - переход на детальную информацию 3.4)
2. Время выхода из зоны ожидания
3. Дата и время включения в ЭО

3.4. Оператор переходит на страничку детальной информации о Клиенте, пример:

<http://server/qms/ui/operator/detailReport?client=a10>

Данные о Клиенте:

1. Номер парковочной карты
2. Количество ротаций (пропусков по причине неявок) ЭО
3. Колонки из 3.2

3.5. Оператор переходит на стартовую страничку 3.2 и выполняет команду меню “Остановить очередь”

3.6. Оператор переходит на стартовую страничку 3.2 и выполняет команду меню “Запустить очередь”

3.7. Оператор переходит на стартовую страничку 3.2 и выполняет команду меню “Выпустить следующего Клиента”

4. Администратор

4.1. Аналогично 3.1

4.2. Аналогично 3.2, если Оператору назначена привилегия “Администратор”, он видит дополнительное меню команд