WebApi платежной системы

1. Краткое описание механизма
2. Базовые определения
   1. Описание сущностей системы
   2. Описание ролей пользователей системы
   3. Описание типов данных используемых в системе
      1. Простых
      2. Составных
3. Аутентификация и авторизация в системе
4. Операции над сущностями
   1. Сообщения
5. Программные функции, реализующие операции над сущностями
   1. Базовые определения
   2. Описание функций
      1. Сообщения

Краткое описание механизма

1. Основным финансовым элементом системы является сущность “Кошелек/ Счет”, все финансовые операции происходят только между экземплярами данной сущности
2. Все серверные действия должны проходить через данную под систему как единую точку входа
3. Единой точкой доступа к WebApi является путь <https://servername/api>
4. Т. к. фактически все запросы к WebApi изменяют состояние системы, то соответственно, по сути, являются POST запросами, содержимое запросов передается в виде JSON объекта превращенного строку в кодировке UTF-8, таким образом, повышается безопасность доступа к системе и организуется единый механизм доступа к WebApi. Поток POST запросов, по сути, выступает просто в качестве канала связи, а не механизма влияния на систему, основную суть будут нести JSON объекты, переданные на необходимые URI

P. S. небольшим количеством GET запросов на выборку данных можно пренебречь с учетом стандартизации и унификации механизма доступа к WebApi

P. S. S. PUT/DELETE запросы не корректно поддерживаются современными браузерами и серверами и требуют дополнительной работы по их реализации соответственно реализовывать их не рационально

Базовые определения

Описание сущностей системы

* Сообщения

Роли пользователей системы

* Пользователь

Описание типов данных используемых в системе

Простых

1. Int: От -2 147 483 648 до 2 147 483 647 32-разрядное целое число со знаком
2. Long: От -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807 64-разрядное целое число со знаком
3. Double: (От ±5,0 × 10^−324 до ±1,7 × 10^308) точность 15-16 значащих цифр
4. Decimal: (От -7,9 x 10^28 до 7,9 x 10^28) / (10^0–28) точность 28-29 значащих цифр
5. UUID: 128-битный номер
6. DateTime: Представляет время, обычно выраженное как дата и время суток
7. String: представляет последовательность, состоящую из нуля или более символов Unicode в кодировке UTF-8

Составных

1. Object: тип данных содержащий любое количество полей других типов данных. Базовое описание: {…}
2. Array: тип данных содержащий любое количество элементов одного типа, простых либо объектов. Базовое описание: […]

**Аутентификация и авторизация в системе**

Данная подсистема использует единый механизм аутентификации и авторизации SimpleMembership

Операции над сущностями

Сообщения

1. Отправка сообщения на сервер
2. Получение списка сообщений с сервера

**Программные функции, реализующие операции над сущностями**

**Базовые определения**

1. Текущее общее количество функций в WebApi – 2 единицы
2. Все программные функции WebApi основываются на [URI](https://ru.wikipedia.org/wiki/URI) - Uniform Resource Identifier
3. Общение клиента с системой идет через POST запросы
4. Данные передаются в виде JSON объектов определенного формата преобразованных в строку символов Unicode в кодировке UTF-8
5. Все URI описанных функций локальны, относительно определенного базового адреса
6. Стандартный шаблон описания функции

**Описание**: операция над сущностью выполняемая данной функцией

**Roles**: Список ролей которым доступна данная функция

**Uri**: адрес функции

**Request**: описание объекта запроса

**Response**: описание объекта ответа

Если в Response указанно базовое описание типа object или array, значит, структура данных полей обьекта Response пока не определена

Базовое описание поля “status”объекта Response: {“code”: ”int”, message: ”string”}

**Описание функций**

**Сообщения**

**Базовый адрес**: <https://servername/api/>messages

Описание: Отправка сообщения на сервер

Roles: Пользователь

Uri: /sendmessage

Request: {“ Message”: “string”}

Response: {“status”: {…}}

**Описание**: Получение определенного количества сообщений

Roles: Пользователь

**Uri**: /getmessages

**Request**: {“CountLastMessages”:”int”}

**Response**: {“status”: {…}, “Messages”: [{…} … {…}]}