# **Протокол взаимодействиЯ Клиентского ПО с СМС центром**

# Структура протокола:

**Структура запроса:**

|  |  |
| --- | --- |
| Метод | POST |
| URI | http://ip:port/ SmppServer/Sms.aspx |
| URI Parameters | с параметрами |
| Body Parameters | с параметрами |
| Content-Type | application/json |

**Параметры URI:**

|  |  |  |
| --- | --- | --- |
| параметр | описание | обязательный |
| key | MD5 хеш сумма получаемая из **пароля**, **зашифрованного сообщения message, номера** **абонента** и **заголовка** СМС.  password + RSA(message) + to + from | да |

**Параметры запроса:**

|  |  |  |
| --- | --- | --- |
| параметр | описание | обязательный |
| message | СМС сообщение отправляемое абоненту | да |
| to | Номер телефона абонента которому будет отправлено СМС сообщение | да |
| from | Заголовок СМС сообщения(От кого пришла СМС) | да |

**Пароль для формирования MD5:**

|  |  |
| --- | --- |
| password |  |

**Пример запроса:**

|  |  |
| --- | --- |
| content-type, request- type | application/json, POST |
| [URL] | http://ip:port/SmppServer/Sms.aspx?key=02be4d5ca2cfe7053ebb1888a8259a19 |
| [BODY] | {      "message": "/aDHOmGWgQxV1v9XWqM1knqN+1N4PXfa+ajFcCSrOkbNxYp7UbVn+Y/eodAdJmFgru1",  "to": "988776611",  "from": "babilon"  } |

**Пример успешного ответа:**

|  |  |
| --- | --- |
| content-type, response-code-type | application/json, 200 |
| [BODY] | {  "success": "СМС сообщение отправлено успешно!"  } |

**Пример НЕ успешного ответа:**

|  |  |
| --- | --- |
| content-type, response-code-type | application/json, 400 |
| [BODY] | {  "error": [Error message]  } |

# Работа с протоколом:

**Шифрования параметра json message:**

Параметр message формируется путём использования метода шифрования RSA. После получения зашифрованной последовательности байтов, данную последовательность необходимо перевести в 64-разрядную кодировку. Ниже представлен код демонстрирующий полный этап шифрования и дешифрования сообщения используя RSA на C#:

|  |
| --- |
| using (RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(2048))  {  string privateKey = "<RSAKeyValue><Modulus>v/JlIYxm3NIKPGI2f/XupN2acxU3LhJom……"  rsa.FromXmlString(privateKey);  string message = "Hello World!";  byte[] bytesMessage = Encoding.UTF8.GetBytes(message);  byte[] encryptByteMessage = rsa.Encrypt(bytesMessage, false);  string encryptX64String = Convert.ToBase64String(encryptByteMessage);  encryptByteMessage = Convert.FromBase64String(encryptX64String);  byte[] decryptByteMessage = rsa.Decrypt(encryptByteMessage, false);  string decryptMessage = Encoding.UTF8.GetString(decryptByteMessage);  } |

Для клиентов(это ваш случай) необходимо иметь при себе только открытый ключ и уметь шифровать сообщения используя открытый ключ RSA. Ниже представлен код на C# который использует только открытый ключ и шифрует сообщение:

|  |
| --- |
| using (RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(2048))  {  string publicKey = "<RSAKeyValue><Modulus>v/JlIYxm3NIKPGI2f/XupN2acxU3LhJom……"  rsa.FromXmlString(publicKey);  string message = "Hello World!";  byte[] bytesMessage = Encoding.UTF8.GetBytes(message);  byte[] encryptByteMessage = rsa.Encrypt(bytesMessage, false);  string encryptX64String = Convert.ToBase64String(encryptByteMessage);  } |

Полное значение открытого ключа можно найти в файле формата xml приложенного вместе с этим протоколом.

**Формирования хеш суммы для параметра key:**

URI параметр key формируется путём высчитывания хеш суммы Md5 из:

* Пароля который описан в "Структура протокола" password
* Зашифрованного параметра message
* Номера абонента кому адресовано СМС сообщение to
* Имени отправителя. От кого пришла СМС from

|  |
| --- |
| key = Md5(password + RSA(message) + to + from) |