Internet Engineering Task Force (IETF)
Request for Comments: 9285
Category: Informational
ISSN: 2070-1721

P. Fältström
Netnod
F. Ljunggren
Kirei
D.W. van Gulik
Webweaving

August 2022

The Base45 Data Encoding

Кодирование данных Base45

Аннотация

Этот документ описывает схему кодирования Base45, основанную на схемах Base64, Base32, Base16.

Статус документа

Документ не относится к категории Internet Standards Track и публикуется лишь для информации.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. Не все одобренные IESG документы являются кандидатами в Internet Standard, см раздел 2 RFC 7841.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке https://www.rfc-editor.org/info/rfc9285.

Оглавление

1. Введение

QR-код применяется для представления текста в форме графического образа. В зависимости от символов текста существуют разные варианты кодирования QR, например, численный (Numeric), алфавитно-цифровой (Alphanumeric), байтовый (Byte). Даже в байтовом режиме типичный считыватель кода QR пытается интерпретировать последовательность байтов как текст в кодировке UTF-8 или ISO/IEC 8859-1. Таким образом, коды QR нельзя применять для непосредственного кодирования произвольных двоичных данных и сначала такие данные нужно преобразовать в подходящий текст. По сравнению с имеющимися схемами кодирования Base64, Base32 и Base16, описанными в [RFC4648], описанная здесь схема Base45 обеспечивает более компактный код QR.

Важным отличием Base45 от других схем является таблица ключей и отказ от дополнения символами =.

2. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе должны интерпретироваться в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

3. Интерпретация кодированных данных

Кодированные данные интерпретируются в соответствии с [RFC4648], но применяется иной алфавит.

4. Кодирование Base45

Возможности сохранения двоичных данных в QR-коде ограничены. На практике двоичные данные представляются символами в соответствии с одним из режимов, определённых в стандартных кодах QR. Постейший режим называется алфавитно-цифровым (см. параграф 7.3.4 и таблицу 2 в [ISO18004]). К сожалению в режиме Alphanumeric используется 45 различных символов, что делает кодировки Base32 и Base64 неэффективными.

Применяется 45-символьное подмножество US-ASCII - 45 символов пригодны для кода QR в режиме Alphanumeric (см. параграф 7.3.4 и таблицу 2 в [ISO18004]). Base45 представляет 2 байта тремя символами, тогда как Base64 представляет 3 байта четырьмя символами.

При кодировании двух байтов [a, b] они **должны** интерпретироваться как число n с базой 256, т. е. как целое число без знака размером 16 битов - n = (a * 256) + b. Это число n конвертируется ы [c, d, e] с базой 45 так, что n = c + (d * 45) + (e * 45 * 45). Отметим, что порядок c, d и e выбран так, чтобы самый левых элемент [c] был наименее значимым. Для значений c, d, e выполняется поиск по таблице 1, дающий 3 строки символов. При декодировании процесс обращается.

При кодировании одного байта [a] он **должен** интерпретироваться как число с базой 256, т. е. как 8-битовое целое число без знака. Это число **должно** преобразовываться в пару чисел с базой 45 [c d] так, что a = c + (45 * d). Для значений c и d выполняется поиск по таблице 1, дающий 2 строки символов.

Строка байтов [a b c d ... x y z] с произвольным содержимым и размером **должна** кодироваться, как описано здесь. Биты слева направо **должны** кодироваться в соответствии с приведённым выше описанием. При чётном числе кодируемых байтов кодированная форма имеет размер кратный 3. При нечётном числе кодируемых байтов последний (правый) байт **должен** кодироваться двумя символами, как указано выше.

При декодировании строк Base45 операции выполняются в обратном порядке.

4.1. Когда Base45 подходит и не подходит

Если двоичные данные предназначены для сохранения в коде QR, предлагаемым механизмом является режим Alphanumeric, использующий 11 битов на 2 символа, как указано в параграфе 7.3.4 [ISO18004]. Индикатор режима расширенной интерпретации канала (Extended Channel Interpretation или ECI) для такого кодирования имеет значение 0010.

Если данные предназначены для передачи иным транспортом, вместо Base45 следует применять соответствующее транспортное кодирование. Например, не рекомендуется сначала использовать Base45, затем кодировать результат с помощью Base64, если данные передаются по электронной почте. В таком случае следует исключить кодирование Base45 и сразу кодировать данные с посощью Base64.

4.2. Используемый в Base45 алфавит

Для режима Alphanumeric определено использование 45 символов, образующих алфавит, как показано в таблице 1.

Таблица 1. Алфавит Base45.

Значение	Символ	Значение	Символ	Значение	Символ	Значение	Символ
00	0	12	C	24	O	36	Пробел
01	1	13	D	25	P	37	$
02	2	14	E	26	Q	38	%
03	3	15	F	27	R	39	*
04	4	16	G	28	S	40	+
05	5	17	H	29	T	41	-
06	6	18	I	30	U	42	.
07	7	19	J	31	V	43	/
08	8	20	K	32	W	44	:
09	9	21	L	33	X		
10	A	22	M	34	Y		
11	B	23	N	35	Z		

4.3. Примеры кодирования

Хотя все представленные примеры показывают кодирование текста, следует отметить, что Base45 подходит для двоичных данных, где каждый октет может иметь любое значение от 0 до 255.

Пример кодирования 1

Строка AB является последовательностью байтов [[65 66]]. Рассматривая все 16 битов, получим 65 * 256 + 66 = 16706. Число 16706 представляется как 11 + (11 * 45) + (8 * 45 * 45), что даёт последовательность чисел с базой 45 [11 11 8]. В соответствии с таблицей 1 получаем кодированную строку BB8.

Таблица 2. Детали примера 1.

AB	Исходная строка
[[65 66]]	Десятичное значение
[16706]	Значение с базой 16
[11 11 8]	Значение с базой 45
BB8	Кодированная строка

Пример кодирования 2

Строка Hello!! в кодировке ASCII является последовательностью байтов [[72 101] [108 108] [111 33] [33]]. Рассматривая блоки по 16 битов, получаем [18533 27756 28449 33]. Отметим, что 33 - последний байт. Преобразование в числа с базой 45 даёт [[38 6 9] [36 31 13] [9 2 14] [33 0]], где последний байт представлен 2 значениями. Кодированная строка "%69 VD92EX0" создаётся поиском по таблице 1. Следует отметить наличие в ней пробела.

Таблица 3. Детали примера 2.

Hello!!	Исходная строка
[[72 101] [108 108] [111 33] [33]]	Десятичное значение
[18533 27756 28449 33]	Значение с базой 16
[[38 6 9] [36 31 13] [9 2 14] [33 0]]	Значение с базой 45
%69 VD92EX0	Кодированная строка

Пример кодирования 3

Строка "base-45" в кодировке ASCII является последовательностью байтов [[98 97] [115 101] [45 52] [53]]. Рассматривая блоки по 16 битов, получаем [25185 29541 11572 53]. Отметим, что 53 - последний байт. Преобразование в числа с базой 45 даёт [[30 19 12] [21 26 14] [7 32 5] [8 1]] , где последний байт представлен 2 значениями. Кодированная строка имеет форму UJCLQE7W581.

Таблица 4. Детали примера 3.

base-45	Исходная строка
[[98 97] [115 101] [45 52] [53]]	Десятичное значение
[25185 29541 11572 53]	Значение с базой 16
[[30 19 12] [21 26 14] [7 32 5] [8 1]]	Значение с базой 45
UJCLQE7W581	Кодированная строка

4.4. Пример декодирования

Пример декодирования 1

Строка QED8WEX0 по результатам поиска в таблице 1 даёт значения [26 14 13 8 32 14 33 0]. Эти числа делятся на блоки по 3, а последний блок содержит 2 числа - [[26 14 13] [8 32 14] [33 0]]. Используя базу 45, получаем числа [26981 29798 33], представляемые байтами [[105 101] [116 102] [33]]. Представление в кодировке ASCII даёт строку "ietf!".

Таблица 5. Детали примера декодирования.

QED8WEX0	Исходная строка
[26 14 13 8 32 14 33 0]	Результаты поиска в таблице
[[26 14 13] [8 32 14] [33 0]]	Группы по 3 значения
[26981 29798 33]	Преобразование с базой 45
[[105 101] [116 102] [33]]	Значения бейтов (база 8)
ietf!	Декодированная строка

5. Взаимодействие с IANA

Этот документ не требует действий IANA.

6. Вопросы безопасности

При реализации кодирования и декодирования важно соблюдать осторожность, чтобы не возникало переполнения буфера и иных проблем. Это включает расчёты с базой 45 и поиск символов в таблице (Таблица 1). Декодер также должен быть устойчив к вводу, включая подобающую обработку любых значений октетов (0-255), в том числе символов NUL (ASCII 0).

Следует отметить, что Base64 и некоторые иные варианты кодирования дополняют стоки и кодирование начинается с одинакового числа символов, тогда как Base45 избегает дополнения. По этой причине следует особенно осторожно кодировать нечётное число октетов, а также нужно соблюдать осторожность при декодировании последовательностей символов, размер которых не кратен 3.

Кодировки Base используют специально сокращённые варианты алфавита для представления двоичных данных. Не включённые в алфавит символы в base-кодированных данных могут возникать в результате повреждения данных или ошибок реализации. Такие символы могут применяться для создания «скрытого канала», по которому не относящиеся к протоколу данные могут передаваться с враждебными целями. Не включённые в алфавит символы могут также передаваться с целью воспользоваться ошибками реализации, ведущими, например, к переполнению буфера.

Реализации **должны** отвергать ввод с недействительным кодированием. Например, они **должны** отвергать ввод (кодированные данные), содержащий не включённые в алфавит символы (Таблица 1), при интерпретации base-кодированных данных.

Хотя строки Base45 содержат лишь символы из таблицы 1, необходимо учитывать некоторые особые случаи. Например, строка FGW представляет число 65535 (FFFF в шестнадцатеричной форме), которое имеет действительное кодирование в 16 битов. Незначительно отличающаяся кодированная строка GGW будет представлять число 65536 (10000 в шестнадцатеричной форме), которое представляется большим, чем 16 числом битов. Реализации **должны** отвергать данные, содержащие триплеты символов, которые при декодировании дают целые числа без знака больше 65535 (FFFF в шестнадцатеричной форме).

Следует отметить, что строка после кодирования Base45 может включать небезопасные для URL символы, поэтому при включении в URL данных Base45 для безопасности URL следует применять %-кодирование.

7. Нормативные документы

[ISO18004] ISO/IEC, "Information technology — Automatic identification and data capture techniques - QR Code bar code symbology specification", ISO/IEC 18004:2015, February 2015, <https://www.iso.org/standard/62021.html>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <https://www.rfc-editor.org/info/rfc4648>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

Благодарности

Авторы благодарны Mark Adler, Anders Ahl, Alan Barrett, Sam Spens Clason, Alfred Fiedler, Tomas Harreveld, Erik Hellman, Joakim Jardenberg, Michael Joost, Erik Kline, Christian Landgren, Anders Lowinger, Mans Nilsson, Jakob Schlyter, Peter Teufl, Gaby Whitehead за их отклики. Спасибо также всем, кто работал долгие годы с Base64 и подтвердил стабильность реализаций.

Адреса авторов

Patrik Fältström
Netnod
Email: paf@netnod.se

Fredrik Ljunggren
Kirei
Email: fredrik@kirei.se

Dirk-Willem van Gulik
Webweaving
Email: dirkx@webweaving.org

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru