Engineer

Gemini

легкий інтернет-протокол для передавання гіпертексту

Що це таке

Gemini — це прикладний протокол для отримання документів, у якому весь обмін складається з одного зашифрованого з'єднання, однієї адреси від клієнта й однієї відповіді від сервера, після чого з'єднання закривається. Власні документи проєкту називають його протоколом Gemini; адреси починаються з `gemini://` , сайт, який через нього віддається, зветься капсулою, а мережа таких сайтів — Geminispace.

Проєкт з'явився в червні 2019 року, і його засновник Solderpunk досі лишається «Benevolent Dictator», тобто доброзичливим диктатором. Специфікацію версії 0.24.1 передано в суспільне надбання за CC0, і вона описує протокол як поступове вдосконалення Gopher, а не як урізаний HTTP. FAQ проєкту дивиться з іншого боку: протокол схожий на щось середнє між HTTP 0.9 і HTTP 1.0, з одним видом запиту, приблизно `GET` .

Клієнт під'єднується до TCP-порту 1965 — це число нагадує про перший пілотований політ за програмою Gemini в березні 1965 року — і виконує рукостискання TLS; обидві сторони мають підтримувати TLS 1.2 або новіший. Потім він надсилає абсолютну URL-адресу, а за нею символи повернення каретки й переведення рядка, і ця адреса може займати щонайбільше 1 024 байти. Сервер відповідає одним рядком заголовка: двоцифровий код статусу, пробіл і короткий текст, значення якого залежить від коду. Коди поділено на шість груп: 10–19 просять ввести дані, 20–29 означають успіх, 30–39 переспрямовують, 40–49 позначають тимчасові збої, 50–59 — постійні, а 60–69 стосуються клієнтських сертифікатів. У разі успіху текстом є MIME-тип, а тіло відповіді йде слідом як сирі байти, без стиснення й без поділу на частини. Наостанок сервер закриває з'єднання повідомленням TLS close_notify, і саме з нього клієнт знає, що отримав усе.

Протокол може передавати файли будь-якого типу, але створювали його разом із власним форматом документів, gemtext, який віддається як `text/gemini`.

Застосуйте поняття до рішення

Застосуйте «Що таке Gemini?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Швидкий сайт, який легко знайти в пошуку і який ваші працівники редагують самі, — без рахунку за кожну зміну в тексті.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Розробила 100 вебзастосунків на HTML/HTML5 — 100 вебзастосунків побудовано

Розробила 100 вебзастосунків на HTML/HTML5 Зв'язатися з нами →

Чому це важливо

Справжнє рішення рідко полягає в тому, щоб перенести сайт на Gemini. За словами FAQ, протокол не має на меті замінити ні веб, ні Gopher, а для публікації того самого вмісту через два чи три з них є власні назви: bihosting і trihosting. Власник сайту вирішує інше: чи варто тримати відчиненими ще одні вхідні двері.

Головне, що дає протокол, — приватність, закладена в саму конструкцію. FAQ прямо каже, що в Gemini немає й ніколи не буде відповідників заголовків Cookie, Referer чи User-Agent. Запит — це URL-адреса й більше нічого, тож серверу майже нічого записувати, навіть якби оператор цього хотів, а в клієнті читача не працює жоден скрипт, який міг би щось передати назад. Серед пріоритетів дизайну FAQ називає «defensive non-extensibility», захисну нерозширюваність: формати запиту й відповіді не мають місця для нового поля заголовка, тож наступна версія не має куди повернути стеження. Для того, хто публікує простий текст, у цьому вся привабливість.

Ціна — значна частина того, чим веб обріс після HTTP 1.0. Cookie немає, тож капсула, якій треба впізнати читача, що повернувся, просить натомість клієнтський сертифікат, а читач мусить його створити й берегти. Центру сертифікації за замовчуванням немає: специфікація наполегливо радить підхід Trust on First Use, або TOFU, за якого клієнт приймає сертифікат, показаний сервером першого разу, і далі закріплює його. Так сертифікат сервера стає довготривалою обіцянкою. Шифрування приховує сторінку, але ім’я сервера передається в розширенні TLS Server Name Indication, і FAQ

визнає, що ім’я хоста досі витікає. Читачам потрібен окремий клієнт Gemini, а не браузер, тож аудиторія невелика. А великі файли підходять погано, бо у відповіді немає ні розміру, ні контрольної суми.

Команді, якій потрібні інтерактивні форми, облікові записи з паролями, аналітика чи багатий макет, місце у вебі. Команда, яка публікує документи й хоче канал без файлів cookie чи клієнтських скриптів, може додати Gemini майже без витрат, якщо вміст уже існує у вигляді файлів.

Як це виглядає на практиці

engineer.company і сам є капсулою за адресою `gemini://engineer.company` . Сторінка Подяки називає Agate програмою, що відповідає на запити Gemini, поряд із Caddy для вебу, Gophernicus для Gopher і Tor для onion-сервісу. Agate бере файли з тієї самої теки, що й три інші програми, тож те саме розгортання, яке оновлює вебсайт, Gopher-нору й onion-сервіс, оновлює й капсулу. Це bihosting із FAQ, розтягнутий на чотири канали й зроблений найдешевшим способом: капсула — лише ще один читач теки, яку кожне розгортання й так наповнює, а не другий сайт, що його треба узгоджувати з першим.

Та сама сторінка чесно говорить про обмеження, описане вище. Вона пише, що для Gemini і Gopher потрібен окремий клієнт, а не браузер, і що Gopher передає відкритим текстом, яку сторінку ви читаєте. Gemini це шифрує, але мережа, з якої ви читаєте, однаково бачить, яку адресу ви запросили. Це той самий витік імені сервера, про який пише FAQ, тільки побачений із крісла читача, а не розробника.

Сторінка Конфіденційність розрізняє журнали звернень застосунків і журнали операційної системи. Застосунки Gemini та Gopher не ведуть журналу звернень; з вересня 2026 року їхні щоденні лічильники зберігають лише загальну кількість запитів. Gopher також має журнал з’єднань операційної системи, який містить адресу джерела; строк зберігання цього журналу — 30 днів.

Де виникають проблеми

Більшість проблем із Gemini видно в першому рядку відповіді, а цей рядок легко прочитати вручну за допомогою самого лише OpenSSL:

```
printf 'gemini://engineer.company/\r\n' |
  openssl s_client -quiet -connect engineer.company:1965 \
    -servername engineer.company
```

Перший рядок, що повертається, — це заголовок, для цієї капсули `20 text/gemini;lang=en` , а за ним іде сторінка.

Постійних читачів попереджають, що капсулу, можливо, атакують. Клієнти, які дотримуються специфікації, зберігають відбиток і дату завершення дії першого побаченого сертифіката. Інший сертифікат до цієї дати вважається можливою ознакою атаки «людина посередині». Оператор, який генерує новий сертифікат під час кожного розгортання або переносить капсулу на нову машину без ключа, викликає це попередження в кожного постійного читача. Додайте ключ і сертифікат до резервних копій сервера, випускайте сертифікат на довгий строк і замінюйте його лише після дати, яку клієнти вже зберегли.

Поле пошуку відповідає статусом 59. Відповідь зі статусом 10 просить читача ввести рядок, і клієнт надсилає його назад як рядок запиту тієї самої URL-адреси, у відсотковому кодуванні. URL-адресу обмежено 1 024 байтами, і довшу сервер мусить відхилити як некоректний запит. Довга відповідь швидко вичерпує цей ліміт, а текст поза ASCII вичерпує його ще швидше: кирилична літера займає два байти в UTF-8 і шість символів після відсоткового кодування. Просіть короткі відповіді, а для чутливого введення, як-от парольної фрази, використовуйте статус 11, щоб клієнт не показував її на екрані.

Капсула забуває читача, щойно той переходить на іншу сторінку. Протокол не має cookie, тож стан сесії, який зберігають по-вебівськи, не переживає переходу від одного запиту до наступного. Капсула, якій треба знати, хто питає, повертає статус 60, клієнт читача створює сертифікат, і сервер упізнає його в наступних запитах. Специфікація обмежує дію цього сертифіката хостом і портом, які його попросили, а також запитаним шляхом і всім, що лежить під ним, тож розміщуйте всі сторінки, яким потрібна ця ідентичність, під одним шляхом. Напишіть на сторінці, для чого потрібен сертифікат, бо читач має на нього погодитися.

Велике завантаження обривається, і цього ніхто не помічає. У відповіді немає ні довжини, ні контрольної суми, тож передавання, що зупинилося на півдорозі, може виглядати завершеним, якщо не зважати на відсутній close_notify. Настанови з найкращих практик радять файли, більші за 100 MiB, віддавати якось інакше. Розміщуйте великі файли на HTTPS і посилайтеся на них із капсули; сторінка Gemini може посилатися на URL-адресу з будь-якою схемою.

Що почитати далі

- Gemini network protocol specification — запити, усі коди статусу, використання TLS і закріплення сертифікатів в одному короткому документі.
- Project Gemini FAQ — цілі проєкту, порівняння з HTTP і Gopher та власна відповідь проєкту на закид, що TOFU слабкий.
- Best practices for Gemini implementations — назви файлів, великі файли, обмеження на переспрямування й налаштування TLS.
- History of Project Gemini — як ідея 2019 року з Gopherspace виросла в окрему мережу, у викладі самого проєкту.

Словник

Обирайте за фактичними умовами

Опишіть робочий процес, наявні інтерфейси та користувачів, яким потрібна зміна. Ми можемо визначити перевірюваний результат і спосіб передачі, який дасть вашій команді змогу підтримувати його.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке Gemini?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/gemini/