oasis

IDEA GENERAL

- 1. RED DESCENTRALIZADA
- 2.SEGURA Y CIFRADA
- 3. FEDERADA
- 4. EMANCIPACIÓN COLECTIVA

INSTALACION

- sudo apt-get install git curl
  curl -sL
  https://deb.nodesource.com/setup_22.x |
  sudo bash -
  sudo apt-get install -y nodejs
  sudo apt-get install npm
  - Instalacion de dependencias
    - git es un programa utilizado para descargar y subir codigo a otras maquinas o servidores
    - El comando instala gitlusado en este caso para descargar el codigo de oasis), y curlbusado para descargar el codigo que instala node, necesario para que oasis funcione)
    - el segundo comando ejecuta un script que configura los repositorios desde los que se descarga node
    - el tercer comando instala node desde los repositorios del comando anterior
- git clone
  https://code.05c8.net/KrakensLab/oasis
  cd oasis
  - este comando utiliza git para descargar una copia del codigo de oasis a tu maquina
- npm install .
  npm run start
  - npm es el gestor de dependencias de node, descarga las cosas que necesita el programa, y lo ejecuta
  - este comando instala las dependencias de node que necesita oasis para funcionar y inicia el programa.

TOUR VIRTUAL POR ARCHIVOS

- DOCS
  - install.md
    - La instalación ahora mismo esta soportada en GNU/Linux
  - code of conduct / contract / contributors / mantainers /
  - security.md
    - Si todo funciona correctamente, es probable que:
      - - Sólo tu computadora puede acceder a Oasis.
      - - Sólo los destinatarios de mensajes privados pueden leer el mensaje.
      - - Sólo tú puedes publicar un mensaje en tu feed.
      - - Sólo HTML básico es soportado en blobs, los cuales no pueden acceder al resto de Oasis.
    - Es importante saber que esto no es una bala de plata:
      - - Tus mensajes públicos pueden ser leídos por cualquiera en la red.
      - - Tus mensajes privados pueden ser leídos por cualquiera que tenga acceso a tu clave privada.
      - - Tu dirección IP puede ser vista por cualquiera que haga peer contigo.
  - CHANGELOGS :
    - - Cambios en las versiones.
- SCRIPTS
  - release.sh
    - Este script automatiza la publicación de una nueva versión del proyecto: crea un branch de release basado en la versión de package.json, lo sube a GitHub, y publica la nueva versión en npm.
  - patch-node-modules.js
    - Este script revisa si el paquete ssb-ref está instalado en node_modules, edita su código fuente para eliminar una función obsoleta y guarda los cambios. Se usa para evitar errores sin esperar una actualización oficial del paquete.
- SRC
  - Backend
    - updater
      - 1. - Ubica los archivos de versión

        Local: ../server/package.json
        Remotos:
        remoteUrl: Repositorio oficial en code.05c8.net.
        remoteUrl2: Espejo en GitHub

        2. - Extrae la versión de un texto JSON con extractVersionFromText(text).

        3- Compara la versión remota con la local con diffVersion(body, callback).

        4. - Intenta obtener la versión remota

        Primero desde remoteUrl (oficial).
        Si falla, intenta con remoteUrl2 (espejo en GitHub).

        5. - Si hay una nueva versión, muestra instrucciones para actualizar.
    - supports
      - 1. -Carga el archivo contacts2.json desde ~/.ssb/flume/ para leer las conexiones del usuario en Scuttlebutt.

        2 - Inicializa la configración desde default.json y permite personalizarla con cli().

        3. - Usa models/main_models para obtener información sobre usuarios (nombre, estado).

        4. -Clasifica contactos en tres listas:

        supports: usuarios seguidos (1).
        blocks: usuarios bloqueados (-1).
        recommends: usuarios recomendados (-2).

        5. -Genera enlaces HTML con los nombres y perfiles de cada usuario.

        6. -Exporta las listas (supporting, blocking, recommending) para usarlas en otras partes del sistema.
    - backend
      - 1. - Carga la configuración desde default.json y permite personalizarla con cli().

        2 -Verifica si Oasis ya está corriendo en el puerto especificado y maneja conflictos (EADDRINUSE).

        3. Usa Koa como servidor web con koaRouter y koaBody para manejar rutas y peticiones.

        4. -Procesa menciones en texto (@usuario) usando ssb-mentions, resolviendo nombres de usuarios.

        5. -Maneja la subida de archivos (BLOBs) a Scuttlebutt (SSB), asegurando que no superen 25MB.

        6. - Devuelve enlaces Markdown según el tipo de archivo (imagen, audio, video o genérico).
  - CONFIG
    - config-manager.js
      - Este script maneja la configuración de Oasis guardada en oasis-config.json.

        Si el archivo no existe, crea uno con valores predeterminados.

        Tiene dos funciones principales: getConfig() para leer la configuración y saveConfig(newConfig) para guardarla.
    - oasis-config.json
      - Este archivo oasis-config.json contiene la configuración de Oasis, con opciones para el tema actual y módulos habilitados.

        Incluye configuraciones para el módulo de billetera (URL, usuario, contraseña y tarifa) y varios módulos activados, como popular, temas, resúmenes, inbox, y más.
    - server-config.json
      - Este archivo server-config.json define la configuración del servidor, incluyendo la programación de replicación, conexiones de red y amigos.

        Configura la replicación automática y parcial, el número de conexiones de amigos y la configuración de "gossip" para la sincronización global.

        También define puertos y protocolos para conexiones entrantes y salientes en varios ámbitos como "public", "device", "tunnel", "onion" y "ws".
  - MODELS
    - main_models.js
      - Gestiona los modelos de datos en OASIS, proporcionando una forma estructurada de acceder a la información de los usuarios y sus interacciones en la red descentralizada de Scuttlebutt.
      - 1.Define y exporta modelos de datos: Proporciona funciones que encapsulan cómo se deben recuperar y estructurar los datos en la aplicación.

        2.Maneja metadatos de usuarios y publicaciones: Extrae información sobre identidades, nombres, imágenes de perfil y descripciones de los usuarios en la red.

        3.Filtra y clasifica mensajes: Define funciones para identificar si un mensaje es público, privado, un post, un comentario, un blog, etc.

        4. Optimiza consultas con cache y estructuras de búsqueda: Mantiene en memoria información sobre nombres de usuarios (all_the_names, feeds_to_name) para agilizar las búsquedas.

        5. Soporta la sincronización de nombres en la red: Usa transpose(LookupTable para consolidar y actualizar los datos de los usuarios a medida que se reciben nuevos mensajes.
  - CLIENT
    - OASIS_CLIENT.js
      - 1.Define una interfaz de línea de comandos (CLI) para configurar y ejecutar Oasis.

        2.Usa yargs para manejar argumentos y opciones del usuario.

        3.Permite configurar opciones como host, puerto, tema, modo público y depuración.

        4.Soporta integración con una billetera ECOin mediante URL, usuario y contraseña.

        5.Carga valores predeterminados desde un archivo de configuración.
    - MIDDELWARE.js
      - 1. Sirve archivos estáticos desde client/assets con koa-static.

        2.-Filtra solicitudes, validando hosts permitidos y bloqueando accesos sospechosos.

        3.Aplica medidas de seguridad como CSP, X-Frame-Options y Referrer-Policy.

        4.Maneja errores mostrando detalles solo si la petición es legítima.

        5.Inicia un servidor HTTP en el host y puerto configurados, permitiendo solo hosts autorizados.
    - GUI.js
      - Este archivo GUI.js gestiona la conexión a un servidor Secure Scuttlebutt (SSB) para Oasis. Intenta conectar usando un socket Unix o TCP y maneja errores si la conexión falla. También permite operar en modo offline y gestiona la apertura y cierre de la conexión de manera controlada.
    - CLI-CMD-ALIASES.js
      - define alias para comandos en una CLI de SSB (Secure Scuttlebutt). Asigna nombres cortos como feed, hist, pub y log a funciones más largas como: createFeedStream, createHistoryStream o getPublicKey, facilitando su uso en la terminal.
    - ASSETS
      - IMAGES
      - STYLES
      - THEMES
      - TRANSLATIONS
  - SERVER
    - SSB_server.js
      - 1. Carga de módulos:

        Usa module-alias para gestionar alias de módulos.
        Importa módulos necesarios para el servidor SSB, como ssb-db, ssb-client, ssb-gossip, etc.

        2. - Configuración:

        Lee la configuración desde un archivo JSON (server-config.json) y los parámetros de la línea de comando.
        Combina la configuración cargada con la proporcionada.

        3. Función de progreso:

        Muestra una barra de progreso para la carga del servidor.

        4. Creación del servidor SSB:

        Usa SecretStack para crear un servidor SSB con varios módulos (gossip, friends, db, etc.).
        Maneja la replicación, la sincronización y otras funciones SSB.

        5. Manejo de la base de datos:

        Redefine el módulo levelup para evitar errores de bloqueo (LOCK).

        6. Cliente SSB:

        Inicializa el cliente SSB con las configuraciones y claves, permitiendo la interacción con el servidor.
        Proporciona un comando de configuración que muestra la configuración del servidor.
    - package-lock.json
      - Es generado automáticamente por npm. Asegura que todas las dependencias se instalen con las mismas versiones exactas en cualquier entorno.

        Contiene un historial detallado de cada paquete y sus subdependencias, evitando inconsistencias.
    - package.json
      - 1. Define la configuración del proyecto Node.js.

        2.Contiene el nombre, versión, autor y licencia del proyecto.

        3.Especifica las dependencias (dependencies) y herramientas para desarrollo (devDependencies).

        4.Contiene los scripts de ejecución (scripts), como start, test, dev, etc.
  - PLUGGINGS-SSB
    - BASICO
      - GESTION DE DATOS
        - SSB-DB
        - SSB-BACKLINKS
        - SSB-QUERY
      - CONEXIONES Y COMUNICACION
        - SSB-CONN
        - SSB-EBT
        - SSB-LAN
        - SSB- ROOM/TUNNEL/CLIENT
        - SSB-UNIX-SOCKET
        - SSB-WS
      - SINCRONIZACION & REPLICACION
        - SSB-REPLICATION-SCHEDULER
        - SSB-ooo (out of order)
      - BUSQUEDA Y ORGANIZACION
        - SSB-TANGLE
        - SSB-SEARCH
      - SEGURIDAD
        - SSB-NO-AUTH
        - SSB-PRIVATE1
        - SSB-ONION
      - RED SOCIAL
        - SSB-ABOUT
        - SSB-FRIENDS
        - SSB-INVITE
      - MULTIMEDIA & ARCHIVOS
        - SSB-BLOBS
        - SSB-MEME
      - INFRASTRUCTURA
        - SSB-MASTER
        - SSB-LOGGIN
        - SSB-PLUGINGS
  - markdown.js
    - Este código convierte texto Markdown en HTML, procesando menciones y generando enlaces a perfiles, hilos, blobs y hashtags en un sistema Secure Scuttlebutt (SSB).
      Utiliza módulos como ssb-markdown, ssb-msgs y ssb-ref. También convierte caracteres de logos en elementos HTML. El resultado es un HTML enriquecido con enlaces y logos personalizados.
  - VIEWS
    - main_views.js (muy resumido)
      - 1. Carga de módulos y dependencias: Importa varias librerías de Node.js (como: fs, crypto, path, lodash, highlight.js y otras) para gestionar archivos, generar contraseñas, manejar configuraciones y formatear datos.
      - 2. Internacionalización (i18n):

        Define un sistema para cambiar el idioma de la interfaz mediante setLanguage.
      - 3. Verificación de actualizaciones:

        Llama a un módulo (updater.js) para comprobar si hay una nueva versión disponible.
      - 4. Generación de vistas HTML: Usa hyperaxe para definir funciones que generan componentes HTML como menús de navegación, listas y enlaces con estilos personalizados.
      - 5. Configuración del menú:

        Construye dinámicamente los enlaces del menú de navegación en función de la configuración (getConfig().modules). Si un módulo está activado (por ejemplo, "popularMod", "inboxMod"), se muestra su respectivo enlace en la UI.
      - 6. Personalización de estilos:

        Permite cargar archivos CSS personalizados si existen en la configuración del usuario.
- install.sh
  - Primero, navega al directorio src/server y muestra un mensaje de bienvenida con la versión del instalador. Luego, ejecuta npm install , para instalar las dependencias de Node.js. Al finalizar, muestra un mensaje indicando que OASIS se ha desplegado correctamente y sugiere ejecutar el script sh oasis.sh para iniciar el servidor.
- oasis.sh
  - Este script oasis.sh automatiza el proceso de iniciar un servidor en un entorno de desarrollo. Primero, navega a la carpeta src/server y ejecuta un servidor SSB (SSB_server.js). Luego, espera a que el servidor esté listo (usando un cliente TCP para verificar la conexión en el puerto 8008). Finalmente, una vez que el servidor esté operativo, navega a la carpeta src/backend y ejecuta el backend (backend.js).
- package-lock.json
- README.md
  - RTFM

CASOS DE USO

- Conectar con personas de confianza , en una situacion de emergencia o en la que no tengas acceso a internet.
- Guardar informacion importante distribuida con la certeza de que ninguna gran compañia podra censurarla
- compartir informacion personal a personas cercanas sin ser utilizado como dato para el capitalismo.
- Poder comunicarte con gente sin tener un algoritmo recopilando datos y dictando que y que no puedes ver y con quien puedes y con quien no puedes comunicarte
- Subir y compartir intercambio de bienes o servicios.

FILOSOFIA

- Anarco pragmatism

Presented with xmind