DevToolsではじめる簡単Nostrプロトコル

あすらも / heguro

- `npub1jw4e8qh6vmyq0n2tkupv7wlfu5h59luk98dcfedf03anh5ek5jkq936u57`
- https://iris.to/heguro@heguro.com

「Nostr勉強会 #0」より

- https://428lab.connpass.com/event/275748/

今回の資料・ソースコード

https://github.com/heguro/nostr-meeting-20230222

PCでご覧のかたは、是非一緒にDevToolsを開いて動かしてみてください！

※YouTubeのページはコンソールのログが流れやすいので、
新しいタブで「about:blank」というURLを開いてからコンソールを開いて
作業することをオススメします！

Nostr プロトコルでの開発がいかに簡単か

始めるのは非常に簡単

なぜか

- すべてがWebSocketで動いてる！！！
  - （WebSocket: ブラウザなどで使えるリアルタイム双方向通信の規格）
- Twitterや各種SNSで定番の「APIキーを取得する」作業すら不要
- 仕様が一カ所にまとまっていて、単純明快
  - https://github.com/nostr-protocol/nips
  - 日本語 → https://scrapbox.io/nostr/NIP

例: ある人の最新の投稿を 10 件取得する

1. WebSocketでリレーサーバーに接続
2. 以下のメッセージを送る

```
[ "REQ", "購読ID", { "pubkey": "その人の公開鍵", "kinds": [1], "limit": 10 } ]
```

3. 投稿内容を含んだメッセージが返ってくる！！

```
[ "EVENT", "購読ID", { <投稿内容> } ]
```

```
[ "EVENT", "購読ID", { <投稿内容> } ]
```

...

```
[ "EOSE", "購読ID" ]
```

ソースコードは `https://github.com/heguro/nostr-meeting-20230222` から

WebSocketのみで書いてみる

```
pubkey = "93ab9382fa66c807cd4bb702cf3be9e52f42ff9629db84e5a97c7b3bd336a4ac"; // @heguroの公開鍵(hex)
subscriptionId = Math.random().toString().slice(2); // 購読IDはランダム

ws = new WebSocket("wss://nostrja-kari.heguro.com"); // 接続して

ws.addEventListener("open", () => { // 接続オープンできたら
  ws.send(JSON.stringify(  // 取得条件を指定してリクエスト
    [ "REQ", subscriptionId, { authors: [pubkey], kinds: [1], limit: 10 } ]
  ));
});
ws.addEventListener("message", (event) => { // メッセージが来たら
  const message = JSON.parse(event.data);
  switch (message[0]) {
    case "EVENT":  // イベント
      if (message[1] === subscriptionId) {
        const event = message[2];
        console.log(new Date(event.created_at * 1000), event.content);
      }
      break;
    case "EOSE":  // End of Stored Events. 投稿を全部返したよ
      if (message[1] === subscriptionId) ws.send(JSON.stringify(["CLOSE", subscriptionId]));
      ws.close();
      break;
    case "NOTICE": console.log("error:", message[1]); break; // エラーなど
  }
})
```

ソースコードは `https://github.com/heguro/nostr-meeting-20230222` から

Nostr 特有の概念: リレーとイベント

- Nostrでの行動はすべて「イベント」で表現される
  - 投稿、リアクション（ふぁぼ・いいね）、フォロー、プロフィール変更...
- リレーサーバーは、基本的には送られてきたイベントを（署名の検証をしつつ）保存し、クライアントに配信するだけ
- クライアントは複数のリレーサーバーに対して、ユーザーの行動に応じたイベントを秘密鍵で署名し配信する

イベントの例

```
{
  "id": "<64文字のhex>",      // 作成時刻・投稿内容から生成される
  "pubkey": "93ab9382fa66c807cd4bb702cf3be9e52f42ff9629db84e5a97c7b3bd336a4ac",
  "created_at": 1676227868,  // イベント作成時刻（＝投稿時刻）
  "kind": 1,                  // 投稿はkind(種類):1
  "tags": [],                 // リプライ先などを指定
  "content": "本文\nだよ〜",   // 改行は`\n`
  "sig": "<128文字のhex>",   // 作成時刻・投稿内容・ID・秘密鍵から生成される
}
```

（投稿イベントにはプロフィール情報が含まれてないので、別にREQしよう！）

ここで問題

既存の投稿だけでなく、リアルタイムに新しい投稿を取得したいときはどうする？

再掲

```
pubkey = "93ab9382fa66c807cd4bb702cf3be9e52f42ff9629db84e5a97c7b3bd336a4ac"; // @heguroの公開鍵(hex)
subscriptionId = Math.random().toString().slice(2); // 購読IDはランダム

ws = new WebSocket("wss://nostrja-kari.heguro.com"); // 接続して

ws.addEventListener("open", () => { // 接続オープンできたら
  ws.send(JSON.stringify(  // 取得条件を指定してリクエスト
    [ "REQ", subscriptionId, { authors: [pubkey], kinds: [1], limit: 10 } ]
  ));
});
ws.addEventListener("message", (event) => { // メッセージが来たら
  const message = JSON.parse(event.data);
  switch (message[0]) {
    case "EVENT":  // イベント
      if (message[1] === subscriptionId) {
        const event = message[2];
        console.log(new Date(event.created_at * 1000), event.content);
      }
      break;
    case "EOSE":  // End of Stored Events. 投稿を全部返したよ
      if (message[1] === subscriptionId) ws.send(JSON.stringify(["CLOSE", subscriptionId]));
      ws.close();
      break;
    case "NOTICE": console.log("error:", message[1]); break; // エラーなど
  }
})
```

ソースコードは `https://github.com/heguro/nostr-meeting-20230222` から

正解は・・・

```
pubkey = "93ab9382fa66c807cd4bb702cf3be9e52f42ff9629db84e5a97c7b3bd336a4ac"; // @heguroの公開鍵(hex)
subscriptionId = Math.random().toString().slice(2); // 購読IDはランダム

ws = new WebSocket("wss://nostrja-kari.heguro.com"); // 接続して

ws.addEventListener("open", () => { // 接続オープンできたら
  ws.send(JSON.stringify(  // 取得条件を指定してリクエスト
    [ "REQ", subscriptionId, { authors: [pubkey], kinds: [1], limit: 10 } ]
  ));
});
ws.addEventListener("message", (event) => { // メッセージが来たら
  const message = JSON.parse(event.data);
  switch (message[0]) {
    case "EVENT":  // イベント
      if (message[1] === subscriptionId) {
        const event = message[2];
        console.log(new Date(event.created_at * 1000), event.content);
      }
      break;
    case "EOSE":  /////  ↓ この2行をコメントアウト ↓  /////
      // if (message[1] === subscriptionId) ws.send(JSON.stringify(["CLOSE", subscriptionId]));
      // ws.close();
      break;
    case "NOTICE": console.log("error:", message[1]); break; // エラーなど
  }
})
```

ソースコードは `https://github.com/heguro/nostr-meeting-20230222` から

これで実行すると・・・

ソースコードは `https://github.com/heguro/nostr-meeting-20230222` から

実際は・・・

- 署名検証の作業が必要
  - 投稿されてからリレーを経由して受信するまでにイベントを改ざんされていないか
- ライブラリーを使うのが手軽
- JavaScriptなら `nostr-tools`
  - https://github.com/nbd-wtf/nostr-tools
  - 取得時自動で署名検証してくれる（手動でもできる）
- DevToolsで読み込む例

```
script = document.createElement("script");
script.src="https://unpkg.com/nostr-tools@1.4.1/lib/nostr.bundle.js";
document.body.append(script);
```

`nostr-tools` ライブラリを使って同等の記述（修正済）

```
pubkey = "93ab9382fa66c807cd4bb702cf3be9e52f42ff9629db84e5a97c7b3bd336a4ac";

relay = NostrTools.relayInit("wss://nostrja-kari.heguro.com");
await relay.connect();  // 接続  （本来はtry-catchで囲むべき）

events = await relay.list([{ // 取得条件を指定して取得開始
  authors: [pubkey], // イベント発行者
  kinds: [1],        // kind1は投稿(ノート)
  limit: 10,         // 過去10個分を取る
}]);
for (const event of events) {  // 取得したイベントを表示
  console.log(
    new Date(event.created_at * 1000), // created_at: 投稿日時 Unix time (秒)
    event.content                      // content: 本文
  );
};
```

ソースコードは `https://github.com/heguro/nostr-meeting-20230222` から

投稿してみる（修正済）

```
pubkey = "93ab9382fa66c807cd4bb702cf3be9e52f42ff9629db84e5a97c7b3bd336a4ac";
privkey = NostrTools.nip19.decode("nsec~~~").data;  // 秘密鍵(hex形式)

relay = NostrTools.relayInit("wss://nostrja-kari.heguro.com");
await relay.connect();  // 接続  (本来はtry-catchで囲むべき)

event = {
  "pubkey": pubkey,
  "created_at": Math.floor(Date.now() / 1000),  // 現在時刻(秒)
  "kind": 1,                // 投稿はkind(種類):1
  "tags": [],               // リプライ先があれば指定
  "content": "本文\nだよ～", // 改行は`\n`
}

event.id = NostrTools.getEventHash(event); // IDを生成
event.sig = NostrTools.signEvent(event, privkey); // sigを生成

pub = relay.publish(event);
pub.on('ok', () => {
  console.log('投稿完了');
});
pub.on('failed', (error) => {
  console.log('投稿失敗:', error);
});
```

ソースコードは `https://github.com/heguro/nostr-meeting-20230222` から

実際にクライアントがやってること

- 複数リレーに同時に接続する
- リンク・画像埋め込みの表示
    - 画像をやりとりする機能はNostrにはないので、外部アップローダーを使う
- リプライ・引用投稿などがあれば元投稿を見にいって表示を置き換える

など・・・

あなたもLet's Nostr!!!!!!!!!!!!!

宣伝

NostrFlu

https://heguro.github.io/nostr-following-list-util/

ソースコードの参考にもどうぞ（かなり雑ですが・・・）

補足

- npub形式の公開鍵があれば、以下のコードでhex形式に変換できます

```
pubkey = NostrTools.nip19.decode(
  "npub1jw4e8qh6vmyq0n2tkupv7wlfu5h59luk98dcfedf03anh5ek5jkq936u57").data;
```

- 実際はNIP-07対応拡張機能（nos2xなど）を使って秘密鍵を直接扱うことなく署名することができ、この方法が**強く推奨**されます
- `nostr-tools` の仕様が素で書くには非常に分かりづらいので、パッケージの仕様や返り値などを確認しながらコーディングできるVS Codeなどのしっかりしたエディターで開発することをオススメします・・・
- スライドで使用したリレーの `nostrja-kari.heguro.com` は現在投稿をホワイトリスト制にしています
    - 別のリレーを使うか、リプいただければリストに入れます

補足2: NIP-07拡張機能（nos2xなど）を使って署名して投稿する
（about:blankでは拡張機能が動かないので、example.comなどで実行）

```
pubkey = await window.nostr.getPublicKey(); // 拡張機能から公開鍵を取得

relay = NostrTools.relayInit("wss://nostrja-kari.heguro.com");
await relay.connect();  // 接続  （本来はtry-catchで囲むべき）

event = {
  "pubkey": pubkey,
  "created_at": Math.floor(Date.now() / 1000),  // 現在時刻(秒)
  "kind": 1,                  // 投稿はkind(種類):1
  "tags": [],                 // リプライ先があれば指定
  "content": "本文\nだよ〜", // 改行は`\n`
}

event.id = NostrTools.getEventHash(event); // IDを生成
event = await window.nostr.signEvent(event); // 拡張機能で署名して、sigを含めたイベントをもらう

pub = relay.publish(event);
pub.on('ok', () => {
  console.log('投稿完了');
});
pub.on('failed', (error) => {
  console.log('投稿失敗:', error);
});
```

ソースコードは `https://github.com/heguro/nostr-meeting-20230222` から