

ソケット通信について

普段、私たちはインターネットを使う際に TCP/IP と呼ぶ通信プロトコルを利用するが、その TCP/IP をプログラムから利用するには、プログラムの世界と TCP/IP の世界を結ぶ特別な出入り口が必要となる。その出入り口となるのがソケットである。

→ 普段利用している HTTP 通信もソケット通信を行っている事が分かる。

ここでのソケットとは TCP/IP のトランスポート層を指す。

そしてそのソケットを利用してアプリケーション層の HTTP 通信などが行われているというイメージである。

簡単にまとめると、

- ・ TCP は、「データの内容はおいといて、通信デバイス間での通信内容を確実に送受信するためのルール」を定めているプロトコル（要するに信頼性）
 - ・ HTTP は、「TCP にさらにルールを追加して、送受信されるデータの形式や送受信タイミングを Web サイト閲覧に最適化する形に定められたルール」
- と考えると分かりやすい。

ソケット通信のライフサイクル

サーバー側

1. **Create** ソケットの作成

エンドポイントを生成するという事

2. **Bind** ソケットを特定の IP アドレスとポートに紐づけ

クライアントがサーバーと接続する際の目印になるという事

3. **Listen** 接続の待ち受けを開始

エンドポイントを監視して、クライアントからの接続を待機するという事

4. **Accept** 接続を受信

接続要求が来たらその要求を受け入れデータの送受信がここで初めて可能になるという事

5. **Close** 接続を切断

データの送受信が終了したらサーバーは接続を切断する。これによりリソースを開放し、他の接続要求を受け入れる準備をする。

クライアント側

1. **create** ソケットの作成

クライアントも最初にエンドポイントを生成する。

2. **bind** ソケットを特定の **IP** アドレスとポートに紐づけ

自分自身の **IP** アドレスとポートを紐づける。しかしこの場合、**OS** が自動で割り振ってくる事が多いためする必要は絶対というわけではない。

3. **connect** リモートソケットに接続

このステップで初めて接続を試みる（相手の **IP** アドレスとポートを指定して）

4. **close** 接続を切断

データの送受信が終了したらクライアントは接続を切断する。

<https://qiita.com/megadreams14/items/32a3eed4661e55419e1c>

<https://envader.plus/article/27>