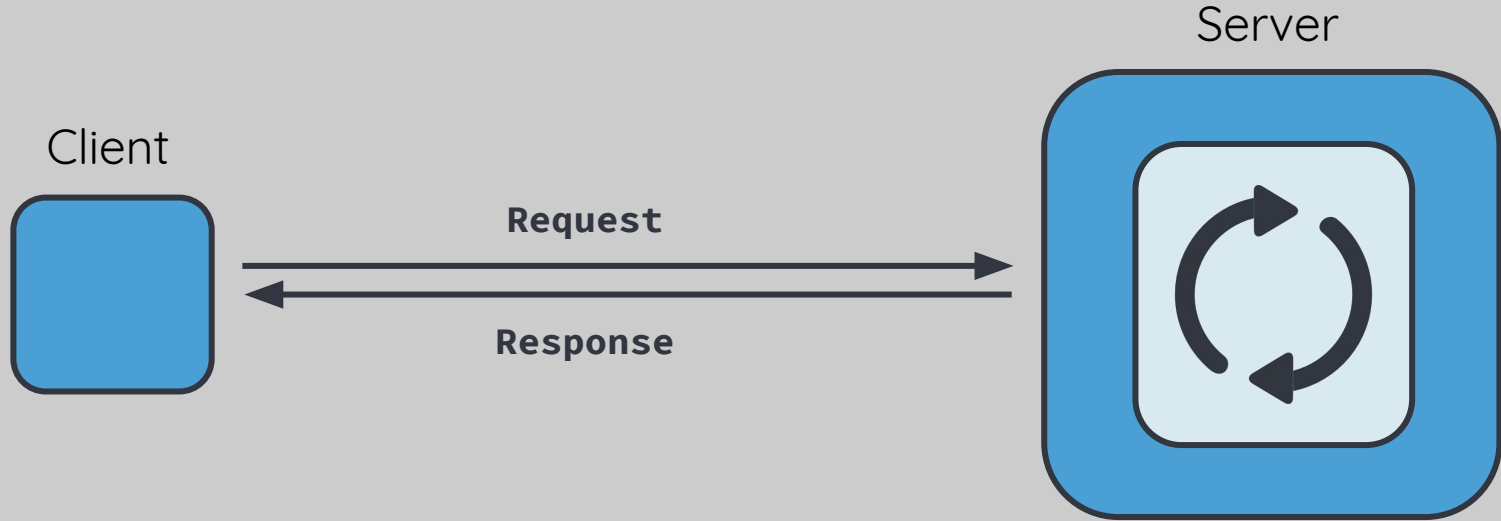


# SERVER

Programming II - Spring 2018



# Client-Server



# HTTP requests

```
Request-Line
* (  general-header |
     request-header |
     entity-header ) CRLF)
CRLF
[message-body]
```

## Example

```
GET www.kth.se/schedule HTTP/1.1
Authentication: <username> <password>
```

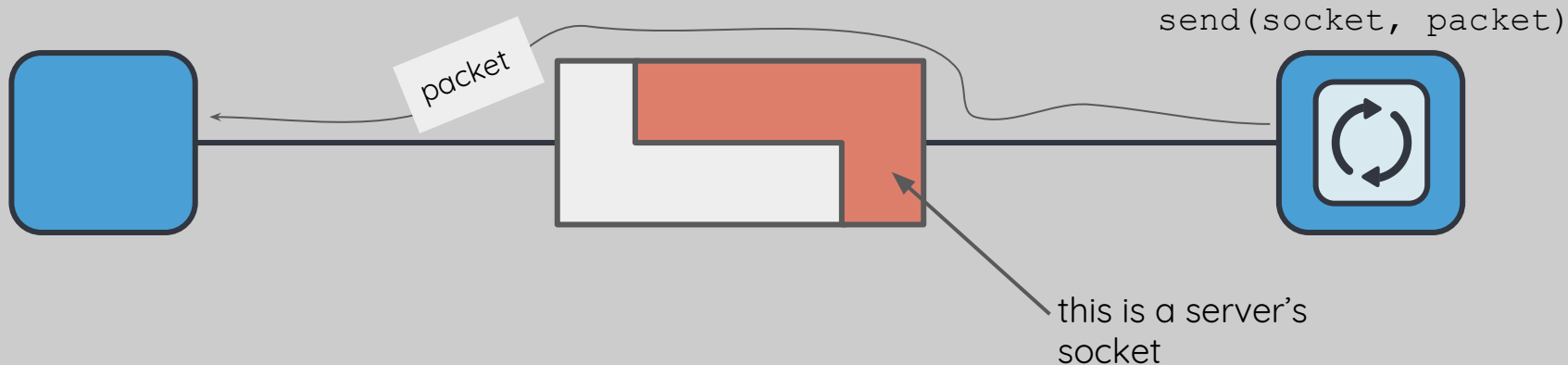
(body not required)

*Note: CRLF is carriage return line feed which basically means new line (`\r\n` in ASCII)*

# What is a socket?

A socket is an **interface** between an application (in our case elixir code) and the network.

Once a **connection** is established with the client, sending data to the socket will send it to the client.



# Socket API

```
# create a listening socket
listen_socket = :gen_tcp.listen(port, opt)

# accept an incoming connection request
client_socket = :gen_tcp.accept(listen_socket)

# receive packet from client
recv = :gen_tcp.recv(client_socket, 0)

# send packet to client
:gen_tcp.send(client_socket, packet)

# close sockets
:gen_tcp.close(client_socket)
:gen_tcp.close(listen_socket)
```

**listen\_socket** is the **listening** socket, used to establish connections.

**client\_socket** is the socket that is used to **send and receive** packets between client and server.

*Good luck!*