

Request Structure Basics

- Mainly 3 parts:
 - Start Line
 - Headers
 - Blank Line
- **Start Line:** Begins with a method (e.g., GET, POST), followed by a path or command, and protocol version.
 - It ends with `\r\n` before the headers start.
 - Example:
 - **HTTP:** GET /index.html HTTP/1.1
 - **FTP:** GET /file.txt
 - **SMTP:** MAIL FROM: <sender@example.com>
- **Headers:** Key-value pairs providing additional information.
 - Each header follows the format `HeaderName: HeaderValue`.
 - Headers are separated by `\r\n`.
 - Terminated by `\r\n` indicating the end of headers and the beginning of an optional message body.
 - Example headers:
 - Host: www.example.com
 - Content-Type: application/json
 - User-Agent: Mozilla/5.0
- **Blank Line:** `\r\n\r\n` separates headers from optional message body.

Message Body

- **Optional:** May contain data being sent by the client.
- **Content-Length:** Specifies length in bytes if present.
- **Chunked Transfer Encoding:** Supports variable-length bodies in some protocols.

Examples

- **FTP:**
 - GET /file.txt
 - Headers: Authentication, Content-Type, etc.
- **SMTP:**
 - MAIL FROM: <sender@example.com>
 - Headers: Authentication, message metadata.

- Complete Example

```
GET /index.html HTTP/1.1\r\n
Host: www.example.com\r\n
User-Agent: Mozilla/5.0\r\n
Accept: text/html\r\n
\r\n
```

Security and Error Handling

- **Authentication:** Methods like tokens, username-password pairs, etc.
- **Security:** Encryption (SSL/TLS), secure authentication protocols (OAuth, JWT).
- **Error Handling:** Status codes (e.g., 200 OK, 404 Not Found) and specific error messages.