

---

## EMAIL SERVICE PROTOCOL

---

### 1. SERVICE LOCATION

---

Host: localhost  
Port: 12345

2. Data Encoding  
Format: JSON (UTF-8).

Validation: All requests/responses must include valid JSON keys as defined below.

3. Transmission Encoding  
Protocol: TCP.

Framing: JSON.

4. Commands & Responses  
All requests include a command field. All responses include a status field.

#### 4.1 Authentication

##### LOGIN

// Request

```
{ "command": "LOGIN", "username": "seb", "password": "secure123" }
```

// Success

```
{ "status": "LOGIN_SUCCESS", "username": "seb" }
```

// Failure

```
{ "status": "LOGIN_FAILURE" }
```

##### REGISTER

// Request

```
{ "command": "REGISTER", "username": "omer", "password": "pass123" }
```

// Success

```
{ "status": "REGISTERED", "username": "omer" }
```

// Failure

```
{ "status": "REGISTER_FAILURE" }
```

#### 4.2 Email Operations

##### SEND

// Request

```
{  
  "command": "SEND",  
  "recipient": "seb",  
  "subject": "Hello",  
  "body": "Meeting at 3 PM."  
}
```

// Success

```
{ "status": "SENT" }
```

// Failure

```
{ "status": "SEND_FAILURE", "error": "Recipient not found" }
```

##### LIST\_INBOX

// Request

```
{ "command": "LIST_INBOX" }
```

```

// Success (with emails)
{
  "status": "INBOX",
  "emails": [
    { "id": 1, "sender": "omer", "subject": "Hello", "timestamp": "2024-05-01T14:30:00Z" }
  ]
}

// Empty
{ "status": "INBOX_EMPTY" }

SEARCH_INBOX
// Request
{ "command": "SEARCH_INBOX", "term": "meeting" }

// Success (matches)
{ "status": "SEARCH_RESULTS", "emails": [...] }

// No matches
{ "status": "NO_MATCHES" }

LIST_SENT
// Request
{ "command": "LIST_SENT" }

// Success
{
  "status": "SENT_LIST",
  "emails": [
    { "id": 2, "recipient": "seb", "subject": "Hello", "timestamp": "2024-05-01T14:35:00Z" }
  ]
}

// Empty
{ "status": "SENT_EMPTY" }

SEARCH_SENT
// Request
{ "command": "SEARCH_SENT", "term": "urgent" }

// Success (matches)
{ "status": "SEARCH_SENT_RESULTS", "emails": [...] }

// No matches
{ "status": "NO_MATCHES" }

READ
// Request
{ "command": "READ", "id": 1 }

// Success
{
  "status": "EMAIL_CONTENT",
  "email": {
    "sender": "omer",
    "recipient": "seb",
    "subject": "Hello",
    "body": "Meeting at 3 PM.",
    "timestamp": "2024-05-01T14:30:00Z"
  }
}

```

```
// Failure
{ "status": "EMAIL_NOT_FOUND" }
```

#### 4.3 Session Controls

##### LOGOUT

```
// Request
{ "command": "LOGOUT" }
```

```
// Response
{ "status": "LOGOUT_SUCCESS" }
```

##### EXIT

```
// Request
{ "command": "EXIT" }
```

```
// Response
{ "status": "GOODBYE" }
```

#### 5. Error Handling

##### Malformed Request

```
{ "status": "ERROR", "error": "Invalid command or format" }
```

#### 6. Security

Password Storage: Hashed using PBKDF2WithHmacSHA256.

#### 7. Sample Exchanges

##### 7.1 LOGIN

###### LOGIN Failure

```
Client → Server: { "command": "LOGIN", "username": "seb", "password": "wrong" }
Server → Client: { "status": "LOGIN_FAILURE" }
```

###### Login Success

```
Client → Server: { "command": "LOGIN", "username": "seb", "password":
"secure123" }
Server → Client: { "status": "LOGIN_SUCCESS", "username": "seb" }
```

##### 7.2 REGISTER

###### Register Success

```
Client → Server: { "command": "REGISTER", "username": "Omer", "password":
"mypassword" }
Server → Client: { "status": "REGISTERED", "username": "Omer" }
```

###### Register Failure (duplicate username)

```
Client → Server: { "command": "REGISTER", "username": "seb", "password":
"anotherpass" }
Server → Client: { "status": "REGISTER_FAILURE", "error": "Username already
exists" }
```

##### 7.3 SEND

###### SEND Success

```
Client → Server: { "command": "SEND", "recipient": "seb", "subject": "Hi",
"body": "Test" }
Server → Client: { "status": "SENT" }
```

###### Send Failure (unknown recipient)

```
Client → Server: { "command": "SEND", "recipient": "unknown", "subject":
"Hello", "body": "Test" }
Server → Client: { "status": "SEND_FAILURE", "error": "Recipient not found" }
```

##### 7.4 LIST\_INBOX

###### Inbox with Emails

```
Client → Server: { "command": "LIST_INBOX" }
Server → Client: {
```

```
    "status": "INBOX",
    "emails": [
      { "id": 1, "sender": "seb", "subject": "Hello", "timestamp": "2024-05-01T14:30:00Z" },
      { "id": 3, "sender": "omer", "subject": "Meeting", "timestamp": "2024-05-02T09:15:00Z" }
    ]
  }
}
```

Inbox Empty

Client → Server: { "command": "LIST\_INBOX" }

Server → Client: { "status": "INBOX\_EMPTY" }

#### 7.5 SEARCH\_INBOX

Search Matches

Client → Server: { "command": "SEARCH\_INBOX", "term": "project" }

Server → Client: {

"status": "SEARCH\_RESULTS",

"emails": [

{ "id": 4, "sender": "seb", "subject": "Projectify Plan", "timestamp":

"2024-05-03T11:00:00Z" }

]

}

Search No Matches

Client → Server: { "command": "SEARCH\_INBOX", "term": "vacation" }

Server → Client: { "status": "NO\_MATCHES" }

#### 7.6 LIST\_SENT

Sent with Emails

Client → Server: { "command": "LIST\_SENT" }

Server → Client: {

"status": "SENT\_LIST",

"emails": [

{ "id": 2, "recipient": "seb", "subject": "Report", "timestamp": "2024-05-

01T16:45:00Z" }

]

}

Sent Empty

Client → Server: { "command": "LIST\_SENT" }

Server → Client: { "status": "SENT\_EMPTY" }

#### 7.7 SEARCH\_SENT

Search Sent Matches

Client → Server: { "command": "SEARCH\_SENT", "term": "report" }

Server → Client: {

"status": "SEARCH\_SENT\_RESULTS",

"emails": [

{ "id": 2, "recipient": "seb", "subject": "Report", "timestamp": "2024-05-

01T16:45:00Z" }

]

}

Search Sent No Matches

Client → Server: { "command": "SEARCH\_SENT", "term": "party" }

Server → Client: { "status": "NO\_MATCHES" }

#### 7.8 READ

Read Success

Client → Server: { "command": "READ", "id": 1 }

Server → Client: {

"status": "EMAIL\_CONTENT",

"email": {

```
    "sender": "omer",
    "recipient": "seb",
    "subject": "Hello",
    "body": "Meeting at 3 PM.",
    "timestamp": "2024-05-01T14:30:00Z"
  }
}
```

#### Read Failure

```
Client → Server: { "command": "READ", "id": 999 }
Server → Client: { "status": "EMAIL_NOT_FOUND", "error": "No email with ID 999" }
```

#### 7.9 LOGOUT

##### Logout Success

```
Client → Server: { "command": "LOGOUT" }
Server → Client: { "status": "LOGOUT_SUCCESS" }
```

#### 7.10 EXIT

##### Exit Goodbye

```
Client → Server: { "command": "EXIT" }
Server → Client: { "status": "GOODBYE" }
```

#### 7.11 Malformed Request

##### Invalid Command

```
Client → Server: { "command": "UNKNOWN", "foo": "bar" }
Server → Client: { "status": "ERROR", "error": "Invalid command or format" }
```