

Responsibilities, services or components? of...

1) Client 2) Carrier/Internet 3) Server

<p>Client = Browser</p> <ul style="list-style-type: none"> Responsibilities, characteristics, features, services, capabilities, and/or components? 	<p>Internet, TCP/IP, HttpRequest, HttpResponse, Https, SSL, ...</p> <ul style="list-style-type: none"> Responsibilities, characteristics, features, services , capabilities, and/or components? 	<p>Server = Web server, Application container, Database server</p> <ul style="list-style-type: none"> Responsibilities, characteristics, features, services , capabilities, and/or components?
--	---	--

Client = Browser and front-end application

- Sends the requests based on link or send button clicks and JavaScript events (AJAX)
- Parses and renders DOM: HTML, CSS, JS
- Downloads and shows images and media
- Provides UI for navigation (links, etc.)
- Possible SPA Routing between “Pages”
- Provides form controls and form submissions (usually HTTP POST)
- UI event handling, like mouse dragging etc.
- Runs JavaScript
- Runs AJAX request and receives responses
- Validation of input (client-side)
- Session management (client-side)
- Saves cookies
- Caching. And Local storage. Debugger.
- Can fetch content from many servers
- Maintains hidden info sent and later again accessed by the server, usually encrypted, (like antiforgerytokens, auth_key, api_key) and so on.
- Support for the carrier security (client-side)

Internet, TCP/IP, HttpRequest, HttpResponse, Https, SSL, ...

- Security: Encryption of data
- Security: We know that the ends are who they say they are
- Routing packages
- Reliability, package handling
- Cache proxies

**MOST LIKELY FEATURES
STILL MISSING FROM ALL
OF THESE LISTS!**

AMAZED?

Server = Web server, Application container, Database server

- Routing, Dispatching
- Hiding the implementation
- Loose coupling of URL to implementing components,
- Restful URIs
- Runs code, maintains object structures
- Interprets and parses requests (incl. AJAX)
- Writes responses e.g. as HTML/JSON/XML
- Validation of input (server-side)
- Session management (server-side)
- Authentication of users **and** client apps **and** requests
- Authorization (user->items, role->actions)
- Conventions/Annotations/Configurations
- Templates helping (Spring Boot, MVC..)
- Database, Database connection, Database user authentication and authorization, (Database encryption)
- Support for the carrier security (server-side)
- Web Services, sockets, ...