

## Funzionamento sito

Il back end dovrà essere fatto in PHP OOP.

Il Front End sarà fatto in Typescript e seguirà la OOP.

Il file index.php sarà l'unico ad essere interrogato.

In base al indirizzamento e al routing, verranno fornite diverse informazioni e dati.

Le 4 tipologie di routing di dominio sono:

chethoo.it		- Sito Default
api.chethoo.it	mail.chethoo.it	- Sottodomini Riservati per servizi
venditore.chethoo.it	superchar.chethoo.it	- pagina annunci e servizi venditore
ciao.ho.sbagliato.chethoo.it		- pagina 404 error

Le tipologie di routing in base alla path dovranno variare in base al tipo di routing di dominio. non è ancora deciso e prestabilito quali saranno, ma per esempio potranno essere:

chethoo.it/profile/

chethoo.it/login/

venditore.chethoo.it/ann/342

## Esempi di Funzionamento

Chethoo.it

Pagina di ricerca

Digitando la URI non vengono implicitamente passati parametri, ne nella path, neppure GET o POST.

Il backend si deve occupare di offrire la versione scarna della pagina, la cosiddetta skeleton.

La versione skeleton della pagina conterrà solo e solamente codice html BASE con layout basilari e primitivi che richiamano le forme e le geometrie del sito.

Verrà inviato anche un file JS, immediatamente da eseguire che conterrà informazioni essenziali per il giusto proseguimento.

Fino a questo punto si è inviato al utente solamente un piccolo file HTML, con poche righe.

Probabilmente allegato un file css che definisce la skeleton, ed insieme un file JS che contiene informazioni necessarie per il caricamento della pagina completa.

Il file JS sarà unico, non ne esisterà una versione per la skeleton ed una per la pagina finale.

Il file JS quindi arrivati a questo punto si occupa di chiedere tramite delle richieste AJAX le parti più pesanti del sito quali le librerie bootstrap, che sono pesanti e verranno caricate dopo.

Successivamente verranno richieste le View, parti della pagina fatte di HTML contenenti già le informazioni e i dati da mostrare.

Infine verranno caricate le immagini.

Per quanto riguarda la pagina di ricerca, le informazioni potrebbero arrivare scaglionate, in particolare non verranno chieste già tutti i modelli e le marche di tutti i veicoli, ma in base alla scelta dei drop down menù , verranno offerti diversi dati.

Il dubbio rimane però se offrire dati in formato JSON oppure direttamente element della select list.

Probabilmente in alcuni casi verranno offerti dati in formato JSON, solo nei casi in cui il caricamento dei dati è dinamico.

Client

BACKEND

Digita  
Chethoo.it

→

Offer la skeleton  
Della pagina di ricercar  
(HTML, CS base, JS completo)

<-

Viene Mostrato nel  
Client la versione skeleton  
Il JS viene eseguito  
Il JS ajax chiede le info della  
Pagina

->

Offre Libreria Bootstrap  
Offre Codice HTML con  
dati statici e dinamici da DB  
Offre CSS completo

<-

Il JS riceve e inietta al  
Posto dello skeleton le view  
Vere.  
Vengono iniettate, con append()  
Sostituite appunto alle skeleton.  
Successivamente chiede Le immagini ->

<-

Offre Immagini e media

Vengono eventualmente richieste altre informazioni  
E altre view dinamiche.  
Per esempio nella pagina di mostra  
Risultati quando si vuole vengono chiesti altri risultati

->

Offre VIEW con dentro i  
dati dei risultati

<-

Se necessario verranno chiesti solo dati  
Da poi inserire manualmente nel value del HTML.  
Come per esempio le select oppure  
Altre tipologie di dati che non necessitano di viste

->

<-

Offre JSON

Il JS manipola il JSON e ne utilizza i dati.

Le richieste HTML alle viste avverranno distinguendo appunto le viste dai dati JSON.  
 In particolare i dati JSON verranno forniti dalla `api.chethoo.it/data/`  
 La versione Skeleton del sito verrà fornita direttamente dal URI di base digitato, come per esempio `Chethoo.it` oppure `venditore.chethoo.it`.  
 Il JS eseguito al suo interno farà le richieste ajax ad `api.chethoo.it/view/nomeView`

Verranno così Distinti i link di ricerca e di utilizzo normale del sito, da quelli necessari per il funzionamento del medesimo.

Un altro esempio

Client	Backend
Digitato <code>venditore.chethoo.it/ann/534</code>	Offerto BOOSTRAP (forse dopo non ora) Offerta la skeleton del singolo annuncio Con html e css base e JS completo
Esecuzione del JS Richiesta della VIEW a <code>GET api.chethoo.it/view/ann/534</code> Il parametro annuncio e il numero 534 viene letto direttamente da URI da JS	Offerto BOOSTRAP (forse prima non ora) Offerto il codice HTML contenente informazioni statistiche della pagina annunci Offerto codice HTML contenente informazioni Dinamiche da DB della pagina annunci (riguardando annunci e l'annuncio 534)
Viene caricato ed eseguito bootstrap. Viene mostrata la VIEW ricevuta sostituendola alla skeleton. vIE Nel caso si abbiano i permessi di modifica viene richiesta alla api la view di modifica <code>api.chethoo.it/view/modann/534</code>	Offerta della view di modifica con al interno i dati precaricati del annuncio 534
Carica la nuova view di modifica sostituendola alla view di visualizzazione dati. In molti casi le form presentano dei controlli di errore, o controlli sul DB. In particolare nel annuncio per esempio non verranno mostrate nella select list tutti i modelli di tutte le macchine, ma solamente i modelli della marca selezionata. Se si seleziona la marca FIAT il client chiede i dati dei modelli delle fiat <code>GET api.chethoo.it/data/marca/FIAT</code>	Offre tutte le informazioni riguardante le FIAT in formato JSON.

<p>Riceve il JSON e adatta la form del annuncio in base ai dati del JSON, modificando appunto le select o i checkbox o il testo.</p> <p>Una volta terminata la modifica del form l'utente clickera su SALVA.</p> <p>I dati relativi al form verranno compressi o sotto forma di POST parameter o sotto forma di JSON e verranno inviati alla API di modifica dati.</p> <p>POST <a href="http://api.chethoo.it/data/ann/534">api.chethoo.it/data/ann/534</a></p> <p>Se l'esito della modifica è andato a buon termine, dopo aver mostrato una icona di caricamento, verrà richiesta la VIEW di visualizzazione dei dati appena modificati Chiedendo appunto</p> <p>GET <a href="http://api.chethoo.it/view/ann/534">api.chethoo.it/view/ann/534</a></p> <p>Questo processo può essere ovviato semplicemente ricaricando la pagina e andando nella VIEW dopo una modifica.</p> <p>Facendo appunto solamente un ricaric pagina su</p> <p><a href="http://Venditore.chethoo.it/ann/534">Venditore.chethoo.it/ann/534</a></p> <p>Tuttavia il caricare la view subito permette di avere un sistema più pulito , senza dover ricaricare bootstrap o eventuali immagini.</p>	<p>Il backend ricevette le informazioni dal frontend le elaborerà, controllerà e le salverà eventualmente nel DB.</p> <p>Viene trasmesso l'esito della operazione al frontend</p> <p>Offerto il codice HTML contenente informazioni statistiche della pagina annunci</p> <p>Offerto codice HTML contenente informazioni Dinamiche da DB della pagina annunci (riguardanti annunci e l'annunci 534)</p>
--	--