# Pattern per la richiesta dati al server

Spesso il client richiede al server dei dati, come ad esempio l’elenco di file destinati ad un Consumer: ognuno di questi file è costituito da molte proprietà, ma il client non ne conosce la struttura (altrimenti si avrebbe duplicazione di codice tra client e server). In questo progetto, quando il client richiede dati al server i quali contengono delle proprietà, esso li invia in forma di mappa, avente per chiave un identificativo e per valore un oggetto con tutte le proprietà di interesse, ad esempio:

**[Client 🡪 Server]** *GET request all’url* /getElencoDocumenti

**[Server 🡪 Client]** *response:*

{

abc5B :

{

nomeDocumento : "fattura idraulico",

dataCaricamento : "11 gennaio 2021",

destinatario : "Luigi Bianchi"

},

1234s :

{

nomeDocumento : "preventivo per Mario",

dataCaricamento : "20 dicembre 2020",

destinatario : "Luigi Bianchi"

}

}

Il client non conosce la struttura dei dati che gli arrivano, quindi a priori non sa come interpretarli, ma può chiederlo al server, ad esempio:

**[Client 🡪 Server]** *GET request:* /getNomePropDestinatarioDocumenti

**[Server 🡪 Client]** *response:*

destinatario

Ricevuta la riposta dal server, il client sa che se vuole elencare i destinatari di tutti i documenti che ha ricevuto, dovrà cercare tra i valori della mappa ricevuta precedentemente la proprietà destinatario ed estrarne il valore.

In questo modo:

* La struttura degli oggetti inviati dal server al client è decisa solamente dal server.
* Non c’è duplicazione di informazioni (e nemmeno di codice) tra client e server (il client non conosce a priori la struttura dei dati, solo che si tratta di una mappa chiave valore).
* Il client possiede già tutte le informazioni sugli oggetti da mostrare e chiede al server solamente come interpretarli, quindi non c’è overhead di dati scambiati tra client e server (il dato completo – cioè la mappa chiave-valore – viene inviato solo una volta).
* Nella mappa data al client, la chiave può essere qualsiasi, l’importante è che sia univoca nella mappa (una mappa non può, per implementazione in JavaScript, avere due *entry* con la stessa chiave).