POST request FormData segítségével - Adatok kezelése Frontend felől a Backend irányába és vissza

A backend fele küldendő FormData attribútumai a formon az input tag-ek *name*-ben megadott azonosítóinevei lesznek, tehát pl.:

<input type="text" id="caller" name="caller" >

Az értékek pedig az input mezők aktuális értékei, amelyeket a felhasználó kitöltött a formon, az alábbi példában a *caller* mező értéke „Brian French”.

Frontend függvény:

//validate ticket data and submit ticket data

    $("#submit").on("click", (evt) => {

    const form = $('form');

    // Trigger HTML5 validity

    let reportValidity = form[0].reportValidity();

    // Then submit if form is OK.

    if(reportValidity){

      let formData = new FormData(document.getElementsByTagName('form')[0]);

      console.log(formData);

      addNewTicket(formData);

    }else{

       evt.preventDefault();

       evt.stopPropagation();

       evt.stopImmediatePropagation();

    }

  });

  function addNewTicket(formData){

myAjax.postAjaxWithFileUpload(apiEndPointCreateTicket, formData, getNewTicketNr);

    nTicketView.resetAllFields();

    }

  function getNewTicketNr(){

 //get the ticket number for the ticket just created and display it

 $.get(apiEndPointTicketNumber,function(data,status){

  $('#ticket-number').text(data.ticketnr);

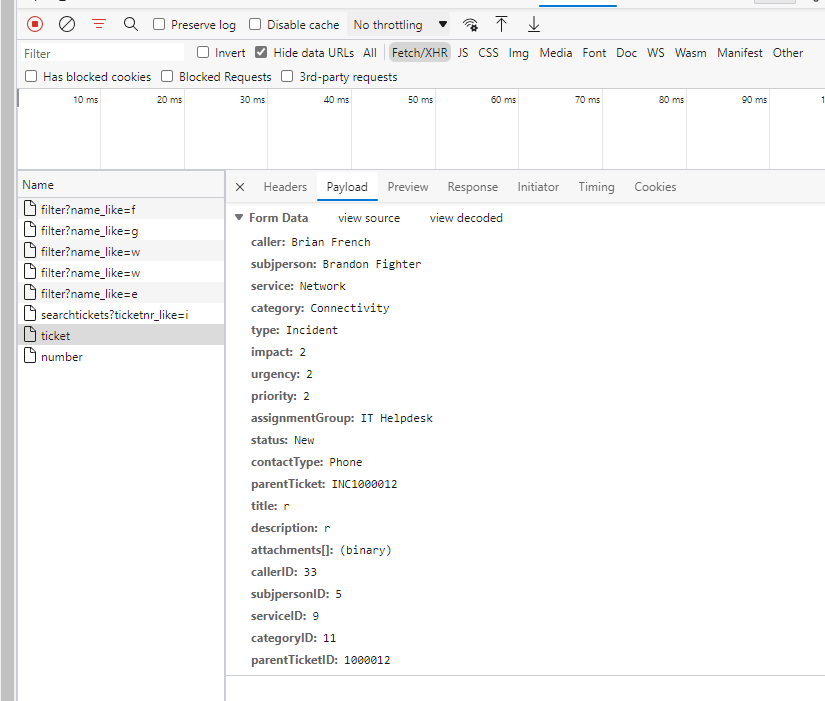
  nTicketView.submissionConfirmationWindow.show();

});

  }

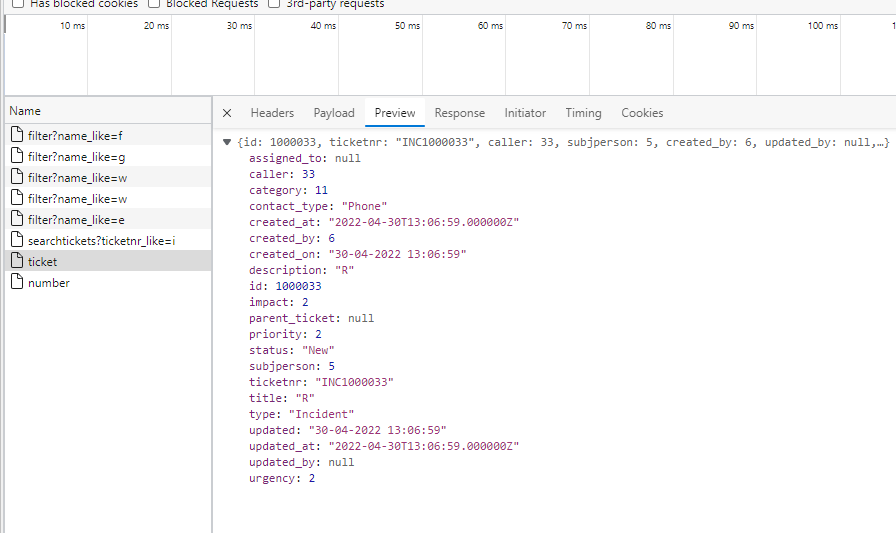
* Ez megy a Backend fele / FormData:

Tehát Laravel controllerben ezek lesznek a beérkező attribútumnevek és értékek, tehát a $request objektumot képező adatok:



* Ez jön a Frontend fele:

Ez lesz a kapott válasz (miután az AB-ba feltöltődtek az új adatok), tehát ezek a Laravel backend-ről a Frontendre érkező attribútumnevek és értékek:



* A backenden történő lekezelés (a $request attribútumai a FormData attribútumaival egyeznek meg):

/\*Új ticket rögzítése, eltárolása AB-ben  \*/

    public function store(Request $request)    {

       //dd($request);

        $ticket = new Ticket();

        $ticket->subjperson = $request->subjpersonID; //Frontend: hidden input!

        $ticket->caller = $request->callerID;  //Frontend: hidden input!

        $ticket->contact\_type = $request->contactType;

        $ticket->status = $request->status;

        $ticket->type = $request->type;

        $ticket->category = $request->categoryID; //Frontend: hidden input!

        $ticket->created\_on = Carbon::now()->format('Y-m-d H:i:s');

        $ticket->updated\_by = $request->updated\_by\_id;

        $ticket->created\_by = Auth::user()->id; //a belogolt user

        $ticket->assigned\_to = null; //első körben ne legyen senkihez assignolva

        $ticket->title = ucfirst($request->title); //első betű nagy betű --> rendezés miatt fontos

        $ticket->description = ucfirst($request->description);

        $ticket->priority = $request->priority;

        $ticket->urgency = $request->urgency;

        $ticket->impact = $request->impact;

        $request->parentTicketID=== null ? $ticket->parent\_ticket = null : $ticket->parent\_ticket = $request->parentTicketID; //Frontend: hidden input!

        $ticket->updated=Carbon::now()->format('Y-m-d H:i:s');

       //sla && time left

        if($request->type =="Request"){

            $prefix='REQ';

        }

        if($request->type =="Incident"){

            $prefix='INC';

              }

        $ticket->save();

       //ticket number mező kitöltése

        $newticket=Ticket::find($ticket->id);

        Ticket::where('id', $newticket->id)->update(['ticketnr' => $prefix.$newticket->id]); //https://stackoverflow.com/questions/35279933/update-table-using-laravel-model

        //attachmentek kezelése

        if(!($request->file('attachments')===null)){

        foreach ($request->file('attachments') as $attachment) {

            $newAtt = new Attachment();

            $newAtt->ticketid =  $newticket->id;

            $newAtt->name = $attachment->getClientOriginalName();

            $newAtt->path = $attachment->store("attachments/". $newticket->id);

            $newAtt->type= $attachment->extension();

            $newAtt->save();

        }

    }

        return Ticket::find($ticket->id);

    }

    /\*A bejelentkezett user által legutoljára rögzített ticket\*/

    public function getLastTicketSubmittedByAuthUser(){

     $ticket = Ticket::where('created\_by','=', Auth::user()->id)->orderBy('created\_on', 'DESC')->firstOrFail();

     return response()->json ($ticket);

    }