U41NK9BM4 : When I got this, then do that :slightly_smiling_face:
U57KYFW67 : oh. I just realized the docs define Task.Task as a type alias of Platform.Task
U57KYFW67 : (it just says type alias Task err ok = Task err ok, which is really confusing before you see it's a cross module reference)
U41NK9BM4 : It's an internal detail.
U57KYFW67 : yeah
U57KYFW67 : that makes more sense now
U41NK9BM4 : I myself looked that and being confused. Ignore for now :slightly_smiling_face:
U4872964V : yeah, it's a problem with the package docs that it doesn't link between packages
U41NK9BM4 : Online book: <http://elmprogramming.com>
U57KYFW67 : or qualify names
U59AF21LJ : Hi! I'm trying to POST some JSON to my own local server running in python with Flask (I have no idea if that's the best idea, especially since I'm not a python fan). The problem is that, even though the server does receive the POST message, the `request.json` is `None` (I print it when I receive it).
```
save : Cmd Msg
save =
  let
    url : String
    url = "update"

    value : Encode.Value
    value =
      Encode.object
        [ ("name", Encode.string "Tom")
        , ("age", <http://Encode.int|Encode.int> 42)
        ]

    body : Http.Body
    body =
      Http.jsonBody value
  in Http.send PostBack (<http://Http.post|Http.post> url body Decode.string)
```
I guess I'm doing something wrong over here. But the header should be fine since I'm using Http.jsonBody.

U41NK9BM4 : It goes well with the official tutorial.
U5XHTBFS6 : <@U59AF21LJ> Is Flask getting the content-type headers?
U5XHTBFS6 : Or you can check on the netwrok tab on your browser's developer tools to see if the requeest headers contain the content=type
U59AF21LJ : <@U5XHTBFS6> Chromes seems to say the type is `xhr`.
U5XHTBFS6 : watdoes it say Content-type: xhr?

U5XHTBFS6 : or is this the request type?
U59AF21LJ : No ok I'm sorry it was listed as xhr, but the request header doesn't have `content-type`.
U59AF21LJ : I'm really not familiar with HTTP stuff so I might be overlooking something silly.
U41NK9BM4 : Are you using Http.Extra?
U26LR8F4H : I'm not entirely sure what I'm looking for here. Basically I extracted a module which performs an http request. <https://gist.github.com/nerdyworm/eb9d4eee5b1e282c8cf2602378772995> . Now I have no idea how to actually use it from my App.elm.
U59AF21LJ : <@U41NK9BM4> No I'm using elm-lang/http
U41NK9BM4 : Ah, I cannot find Http.jsonBody.
U59AF21LJ : <http://package.elm-lang.org/packages/elm-lang/http/1.0.0/Http#jsonBody>
U5XHTBFS6 : <@U59AF21LJ> Flask parses the body based on the Content-type header. If it's not application/json, it will probablyu assume it's form-encoded and the .json property will be none
U5XHTBFS6 : I'm very new to Elm, so there might be a better way, but what I do is usee Http.request and pass the content type explicitly.
U41NK9BM4 : <@U59AF21LJ> Got it!
U59AF21LJ : <@U5XHTBFS6> Yeah it seems to be the case, but I don't understand as the doc specifically says it adds the content-type header.
U5XHTBFS6 : ``` let

```
            url =
                "<http://localhost:5000/>"

            request =
                Http.request
                    { method = "POST"
                    , headers =
                        [ Http.header "Content-type" "application/json" ]
                    , url = url
                    , body = body
                    , expect = Http.expectString
                    , timeout = Nothing
                    , withCredentials = True
                    }
        in
            ( model, Http.send GetSpotList request )
```

U5XHTBFS6 : This is how I'm doing it^
U5WD40ZA9 : Anyone know a great package for file uploading in elm?
U48AEBJQ3 : <@U26LR8F4H> There are a couple of paths to take, but I think that in your case, I would get rid of `type Results` and change the function to `shorten : (Result Http.Error Bitly -&gt; msg) -&gt; String -&gt; Cmd msg`
U41NK9BM4 : <@U5XHTBFS6> Why are you doing a GET?
U5XHTBFS6 : it should be a POST, that was a copy paste error :smiley:
U41NK9BM4 : OK. Makes more sense now
U5XHTBFS6 : There, fixed it. Thanks for spotting the error
U26LR8F4H : <@U48AEBJQ3> thanks, worked like a charm :slightly_smiling_face:
U2UUNG9KK : Thank you <@U41NK9BM4>