آموزش متصل کردن phoenix به درگاه واسط زرین پال

نویسنده : ترانگل
https://trangell.com

زرین پال یکی از درگاه های واسط فعال در ایران می باشد که اصل این چنین درگاه های واسط میسر کردن پرداختی کاربران کسب کار های کوچک می باشد . به این دلیلی که این کسب و کار ها نمی توانند به علت محدودیت های ثبت شرکت یا مشخص کردن مکان فیزیکی به صورت مستقیم از بانک مورد نظرشان درگاه پرداخت بگیرند.

قبل از هرکاری فایل mix خودتان رو به این دو کتابخانه به روز رسانی نمایید

```
{:elixir_xml_to_map, "~> 0.1.2"},
{:soap, "~> 0.1.2", github: “shahryarjb/soap”}
```

اگر برای شما سوال شده که چرا من دارم از گیت هاب خودم به روز رسانی رو انجام می دم . به این دلیلی می باشد که کتابخانه مذکور یعنی soap با آخرین نسخه های httpoison مشکل دارد و همین مورد باعث می شود نتوانید کتابخانه را نصب کنید بنده فقط نسخه httpoison را بالاتر برده ام .

بعد از آن نوبت معرفی کردن این دو کتابخانه در application می باشد به صورت مثال

```
extra_applications:
[:logger, :runtime_tools, :httpoison, :elixir_xml_to_map, :soap
]
```

متاسفانه این کتابخنه یکمی مشکل دارد و شما باید در فایل کانفیگ خودتان آن را معرفی کنید

```
config :soap, :globals,
    version: 1.2
```

حال شما می توانید به سادگی با دستور mix deps.get این دو کتابخانه را نصب کنید و مراحل بعدی که اتصال به زرین پال می باشد را انجام بدهید.

نکته : قبل از هرچیزی اطلاعات وارد شده از طرف بنده در این پست اطلاعات مربوط به درگاه تست می باشد و شما می توانید با تغییر لینک تست به لینک اصلی به صورت کامل بدون هیچ تغییر دیگری از آن استفاده کنید

مرحله اول:
قبل از هرچیزی دو تابع ارسال و callback را می سازیم که به شرح زیر می باشند

```
 #send fun
  def zarinpal_send(amount) do
    wsdl_url = "https://sandbox.zarinpal.com/pg/services/WebGate/wsdl?WSDL"
    operation = "PaymentRequest"
    merchantid = "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX"
    description = "this is test"
    email = "m@m.com"
    mobile = "09361111111"
    callbackurl = "http://localhost:9991/zarinpal/callback"

    params = %{MerchantID: merchantid, Amount: "#{amount}", Description: description,
Email: email, Mobile: mobile, CallbackURL: callbackurl}

    with {:ok, wsdl} <- Soap.init_model(wsdl_url, :url),
      {:ok, %Soap.Response{body: body, headers: _headers, request_url: _url, status_code:
_code}}  <- Soap.call(wsdl, operation, params) do
        convert_zarinpal_xml(body)
    else
      nil ->      {:unknown_error}
      _   ->      {:unknown_error}
    end
  end
```

در این تابع من فقط اون پارامتر هایی که نیاز می باشد و اطلاعات زرین پال خودم را ارسال کردم نه چیز دیگیری و همینطور از چک کردم ببنیم آیا مشکلی در این وسط برای اتصال به زرین پال وجود دارد یا خیر؟
و همینطور در تابع callback

```
def zarinpal_callback(authority, amount) do
  wsdl_url = "https://sandbox.zarinpal.com/pg/services/WebGate/wsdl"
  merchantid = "XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX";
  operation = "PaymentVerification"
  params = %{MerchantID: merchantid, Authority: authority, Amount: amount}
```

```
  with {:ok, wsdl} <- Soap.init_model(wsdl_url, :url),
    {:ok, %Soap.Response{body: body, headers: _headers, request_url: _url, status_code:
_code}}  <- Soap.call(wsdl, operation, params) do

      convert_zarinpal_call_back_xml(body)
  else
    nil ->      {:unknown_error}
    _   ->      {:unknown_error}
  end

end
```

تا اینجا ما چیز زیادی نداشتیم فقط خواستیم یک درخواستی را به زرین پال بفرستیم و یک جوابی بگیریم .

مرحله دوم:

جواب های دریافتی هیچ خروجی تا به حال نداشتند و حال زمان این است که بیاییم مواردی که نیاز داریم را دریافت کنیم به عنوان مثال کد تراکنش یا ... که من در اینجا دو تابع درست کردم یکی برای تابع ارسال و دیگیری برای تابع برگشت

```
def convert_zarinpal_xml(body) do
    body
    |> XmlToMap.naive_map
    |> Map.get("{http://www.w3.org/2003/05/soap-envelope}Envelope")
    |> Map.get("{http://www.w3.org/2003/05/soap-envelope}Body")
    |> Map.get("{http://zarinpal.com/}PaymentRequestResponse")
  end

  def convert_zarinpal_call_back_xml(body) do
    body
    |>  XmlToMap.naive_map
    |> Map.get("{http://www.w3.org/2003/05/soap-envelope}Envelope")
    |> Map.get("{http://www.w3.org/2003/05/soap-envelope}Body")
    |> Map.get("{http://zarinpal.com/}PaymentVerificationResponse")
  end
```

همانطور که می بینید بنده در این دو تابع xml رو به یک map تغییر کاربری دادم که به عنوان خروجی های توابع بالا از اون ها استفاده کنم.

توجه: بنده با مطالعه ارور های برگشتی و همینطور درخواست هایی که از بانک های دیگیری که قبلا کار کردم گرفتم, یک مجموعه اروری رو درست کردم که دیگیر مجبور به انجام دوباره آن نباشید . لطفا این ارور ها را در جایی قرار بدهیم

https://gist.github.com/shahryarjb/1ced23bbbffd5bdadc5d6652791537b8

مرحله سوم:
حالا زمان این هست که توابع اکشن خودمان را بسازیم . بدهی می باشد که شما از قبل روتر و ویو مربوط به صفحه فرم ارسال قیمت را نوشته اید و این آموزش در ساده ترین شکل درست شده است و می تواند مراحل به صورت دیگیری باشد

```
  def check_pay(conn, %{"amount" => amount}) do
    conn = put_session(conn, :amount, amount)
    case ZarinPal.zarinpal_send(amount) do
      nil ->
        check_status_send(conn, {:unknown_error}, :unknown_error)

      {:unknown_error} ->
        check_status_send(conn, {:unknown_error}, :unknown_error)

      %{"{http://zarinpal.com/}Authority" => authority, "{http://zarinpal.com/}Status" =>
status}  ->
        check_status_send(conn, status, authority)
    end
  end
```

به عنوان مثال من در تابع get خودم قیمتی رو از کاربر دریافت می کنم و بعد از آن قیمت را سشن کرده و بعد به تابعی که در بالا نوشته ایم یعنی تابع send مون می فرستم و اون هم در جواب خروجی هایی رو برای شما فراهم می کنه که باز در یک case اون رو بررسی می کنم و در صورت درست بودم که سومین احتمال من هست به تابعی به شرح زیر می فرستم

```
  def check_status_send(conn, "100", authority) do
      redirect conn, external: "https://sandbox.zarinpal.com/pg/StartPay/#{authority}/
ZarinGate";
  end

  def check_status_send(conn, status, _) do
    {:error, error} = ZarinPal.zarinpal_status(status)

    conn
    |> fetch_session
    |> delete_session(:amount)
    |> put_flash(:error, error)
```

```
  |> redirect(to: "/plans")
 end
```

تـابـع بسیار سـاده هسـت و فکر نمی کنم نیازی بـه تـوضیح خـاصی داشـته بـاشـه فـقط اگـر درسـت بـود که کاربـر رو ریدایرکت می کنه به سایت زرین پال و اگر نبود که نمایش ارور مورد نظر رو انجام می ده
تا اینجا شما باید یک بررسی کوچیک بکنید و ببنید آیا به درگاه تست زرین پال هدایت می شوید یا خیر؟
اگر مشکلی نباشد زمان ای هست که بیاییم تابع برگشت را بنویسیم:

```
 # call back

 def check_pay_back(conn, %{"Authority" => authority, "Status" => "OK"}) do
  case ZarinPal.zarinpal_callback(authority, get_session(conn, :amount)) do
   nil ->
    check_status_back(conn, {:unknown_error}, :unknown_error)

   {:unknown_error} ->
    check_status_back(conn, {:unknown_error}, :unknown_error)

   %{"{http://zarinpal.com/}RefID" => refid, "{http://zarinpal.com/}Status" =>  status}  ->
    check_status_back(conn, status, refid)

  end
 end

 # پرداخت کنسل شده
 def check_pay_back(conn, %{"Authority" => _authority, "Status" => "NOK"}) do
  if get_session(conn, :amount) != nil do
   check_status_back(conn, "-17", "")
  else
   {:error, error} = ZarinPal.zarinpal_status("-1")
   conn
   |> fetch_session
   |> delete_session(:amount)
   |> put_flash(:error, error)
   |> redirect(to: "/plans")
  end
 end

 def check_pay_back(conn, _params) do
  {:error, error} = ZarinPal.zarinpal_status("-1")
```

```
    conn
    |> fetch_session
    |> delete_session(:amount)
    |> put_flash(:error, error)
    |> redirect(to: "/plans")
  end


  defp check_status_back(conn, "100", refid) do
    {:error, error} = ZarinPal.zarinpal_status("100")
    conn
    |> fetch_session
    |> delete_session(:amount)
    |> put_flash(:success, "#{error} شماره تراکنش شما: #{refid}")
    |> redirect(to: "/plans")
  end


  defp check_status_back(conn, status, _refid) do
    {:error, error} = ZarinPal.zarinpal_status(status)

    conn
    |> fetch_session
    |> delete_session(:amount)
    |> put_flash(:error, error)
    |> redirect(to: "/plans")
  end
```

شرح روتر

```
conn
    get "/plans" , PayController, :plans
    post "/zarinpal/sendpay" , PayController, :check_pay
    get "/zarinpal/callback" , PayController, :check_pay_back
```

- لینک اول صفحه ای می باشد که می توان فرم پرداخت را درست کرد
- لینک دوم صفحه ارسال به زرین پال می باشد
- لینک سوم صفحه برگشت از زرین پال می باشد

شما در حقیقت یک بار مبلغ را ارسال کرده و یک بار هم آن را ورفای کردید و در صورت نبود مشکل آن را به کاربر نشان دادید. لازم به ذکر هست که شما می توانید مراحب مثل سشن کرد و ... را تغییر بدهید . این آموزش بر

اساس یک سشن ساده قیمت بوده که مقایسه می کرده است شما می توانید با پیش فاکتور یا ... بررسی بهتری را بسازید.

امید وارم مفید واقع شده باشد
با تشکر از مجتبی ناصری برای ساخت نسخه پایه اتصال

به روز رسانی
تیم سازنده افزونه سوآپ پول منو تایید کرده می تونید از نسخه خودش نصب کنید البته باید از برنچ مستر دانلود کنید

https://github.com/potok-digital/soap/pull/44