


# آموزش اتصال phoenix به درگاه واسط zarinpal به وسیله json api

 devheroes.club/t/phoenix-zarinpal-json-api/3884/1

با درود خدمت دوستان عزیز و اساتید محترم. همانطور که در جریان هستید بیشتر درگاه های رسمی کشور با سوپا نوشته شده اند ولی درگاه های واسط به مرور در حال پیاده سازی api های کارآمدتر و ساده تر هستند .

قبلا روش اتصال به زرین پال را در لینک زیر به وسیله سوپا آموزش دادیم

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,}:  
github: "shahryarjb/soap", "0.1.2", {:"soap", "0.1.2"} اگر برای شما سوال شده که چرا من دارم از گی...

خیلی از مباحث در لینک بالا مطرح شده است فقط در این آموزش معرفی چند لینک و یکی دوتا فانکشن ساده و راه ارتباط به وسیله json را انجام خواهیم داد.

توجه داشته باشید شما بر اساس نیازمندی های خودتان می توانید خطا های بیشتری را پیاده سازی کنید فقط کافیست داکيومنت وب سایت مذکور و همینطور خروجی دریافتی از این وب سایت را دریافت کردن و تبدیل به یک پترن سفارشی دلخواه خود نمایید

## مرحله اول

در این مرحله من روی پروژه خودم پلاگین httpoison را نصب کردم که می توانید در وب سایت hex.pm ان را روی پروژه خودتان اضافه کنید

به صورت مثال

```
defp deps do
  ]
  ,{"phoenix", "~> 1.4.0:}
  .....
  ....
  ,{"httpoison", "~> 1.5:}
```

و همینطور آن را در بخش زیر نیز معرفی کردم

```
def application do
  ]
  ,{[] ,mod: {BankError.Application
[extra_applications: [:logger, :runtime_tools, :httpoison
[
end
```

## مرحله دوم

در این مرحله یک فایل در پروژه خودمان می‌سازیم که های زیر در آن قرار می‌گیرد

<https://gist.github.com/shahryarjb/ed055433a652bba1a081e0c1d1bc982a>

zarinpal\_json\_api.ex

```
defmodule BankError.Extera.BankProvider.Zarinpal do

  def get_token(type) do
    !sms_body = Jason.encode
  }%
  , "MerchantID" => "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
  , "Amount" => "1000"
  , "CallbackURL" => "https://YORSITE.com/back-pay/zarinpal"
  Description" => type"
  {
```

This file has been truncated. show original

لازم به ذکر هست این کد ها تکرار در آن ها زیاد هست و دلیل آن توانایی درک بهتر در این آموزش می باشد شما می توانید کد های کمتر و تمیز تر و داینامیک تری را در پروژه خود ایجاد کنید

## مرحله سوم

شما در این مرحله سه تابع دارید که هر کدام به ترتیب باید کار هایی را برای شما انجام بدهند

۱. در این تابع یعنی `get_token` شما یک بار به زرین پال درخواست می دهید و مواردی مثل

MerchantID کد درگاه شما در اینجا تست می باشد  
Amount قیمت محصول شما برای بررسی و ولید سازی مهم است  
CallbackURL لینک بعد از پرداخت یا عدم پرداخت باید برگشت کند به سایت شما  
Description توضیحات پرداخت

به همین سادگی شما در این مرحله که کد های آن بالا قرار دارد و به وسیله HTTPoison یک درخواست `post` می زنید و توکن برای شروع انتقال کاربر به درگاه بانکی را دریافت می کنید

## مرحله چهارم

شما در این بخش در کنترل پروژه فونیکس خودتان می آید و کد دریافتی از زرین پال را با همراه کاربر ریدارکت می کنید به درگاه زرین پال

به صورت زیر

```
("{redirect(external: "https://sandbox.zarinpal.com/pg/StartPay/#{authority <|
```

conn

## آخرین مرحله

حال شما باید اعتبار سنجی فرمایید که آیا کد تراکنش کاربر درست می باشد و توکن آن از قبل ولید بوده یا خیر

که تابع زیر می باشد

```
def validate(zarinpal_token) do
  )!sms_body = Jason.encode
  }%
  ,"MerchantID" => "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
  ,"Amount" => "1000"
  "{Authority" => "#{zarinpal_token}"
  {
    (
      send_info = HTTPoison.post
      ,""https://sandbox.zarinpal.com/pg/rest/WebGate/PaymentVerification.json
      , sms_body
      ]
    ,{"Content-Type", "application/json"}
    [
      (zarinpal_status(send_info
      end
    end
```

در این بخش شما توکن و همینطور توکن بعد از خرید را ارسال کرده و در صورت اوکی بودن درخواست ثبت در دیتابیس خود می نمایید

نمونه ورودی تابع اکشن در کنترل در موقع برگشت از زرین پال

```
def back_pay(conn, %{"Authority" => authority, "Status" => "OK", "provider" => "zarinpal"})
do
```

موفق باشید