



راهنمای نحوه اتصال به درگاه پرداخت اینترنتی رای پی

شهریور ۱۴۰۰

مقدمه

این راهنما نحوه اتصال برنامه نویسان به درگاه های پرداخت از طریق سرویس های پرداخت رای پی را توضیح میدهد. پرداخت یار رای پی از وب سرویس Rest (Web API) برای برقراری خدمات پرداخت استفاده نموده است. همچنین بدنه درخواست ها و پاسخ ها با فرمت Json می باشد.

پیش نیازها

برای استفاده از این سرویس و اتصال به درگاه پرداخت، لازم است ابتدا از آدرس <https://raypay.ir> اقدام به ثبت نام نمایید.

بعد از تایید درخواست فوق، اطلاعات زیر برای کسب و کار صادر می شود که از بخش لیست کسب و کارها در پنل رای پی قابل دسترسی است:

- شناسه کاربری (User ID)
- شناسه کسب و کار (Marketing ID)

شرح مراحل پرداخت

قبل از شروع پیاده سازی بهتر است یک بار فرایند پرداخت را با هم مرور کنیم.

- **مرحله اول:** ارسال اطلاعات تراکنش و دریافت توکن پرداخت
- **مرحله دوم:** انتقال کاربر به صفحه پرداخت با استفاده از توکنی که از مرحله اول دریافت شده است.
- **مرحله سوم:** بعد از فرایند پرداخت، کاربر به آدرس بازگشتی که در مرحله اول ارسال شده، برمیگردد. در این مرحله برای نهایی شدن تراکنش، باید متد **verify** فراخوانی شده و وضعیت نهایی تراکنش دریافت شود.

مرحله اول : ارسال اطلاعات و دریافت توکن

ابتدا باید پارامترهای موجود در جدول زیر رو با متد POST به آدرسی که مشخص شده ارسال کنید. به نوع داده ها ، نام فیلد و اجباری بودن یا اختیاری بودن آن ها توجه کنید.

method: POST

URL: <https://api.raypay.ir/raypay/api/v1/Payment/Pay>

پارامترهای ورودی

پارامتر	نوع	وضعیت	توضیحات
amount	string	ضروری	مبلغ تراکنش به ریال . مقدار این پارامتر میتواند از 1,000 ریال تا 500,000,000 ریال باشد.
invoiceID	string	ضروری	شناسه پرداخت یا شناسه ارجاع بانکی رای پی که در هر تراکنش باید به صورت یکتا تولید شود.
userID	string	ضروری	شناسه کاربری که در بخش لیست کسب و کارها در پنل رای پی قابل دریافت می باشد.
marketingID	string	ضروری	شناسه کسب و کار که در بخش لیست کسب و کارها در پنل رای پی قابل دریافت می باشد.
redirectUrl	string	ضروری	آدرس بازگشت به سایت پذیرنده پس از فرآیند پرداخت (این آدرس باید حتما با http:// یا https:// شروع شود).
factorNumber	string	اختیاری	شماره فاکتور یا شماره سفارش جهت شناسایی درخواست هنگام بازگشت از درگاه

خروجی متد

```
{
  "Data":
    "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJBY2Nlc3N0b2t1biI6IiNtOG9PNElUS3FzMWw0Ym1GL1dybTU0dT10TnAwZDU5aklzMGhCcmZlQ2s9IiwiaGVhZGFubWVudHMiOiIiLCJFbWVpbGl6IiIsIk1vYm1sZSI6IiIsIlJlcXVlc3REYXRlIjoiiwiUmVxdWVzdFRpbWUiOiIiLCJXcm10aGVhZGVySUQiOiIiLCJSZWRCcmVjdFVybCI6IiIsIkZhY3Rvcnk51bWJlciI6IiIsIm5iZiI6MTYzMTUxOTQyOQwiZXhwIjoxNjMxNTMzODI0LCJpcyYXQiOjE2MzE1MTk0MjgsIm1zcyI6IiNhbWluclucF5LklDVC5DTyIsImF1ZCI6IiNhbWluUGF5bWVudFVzZXJzIn0.Fs3pEB0Is3V3HWsngyDU071UXLh7Rh6yZD0gw5tDP1g",
    "IsSuccess": true,
    "StatusCode": 0,
    "Message": "عمليات با موفقیت انجام شد"
}
```

```
{
  "IsSuccess": false,
  "StatusCode": 2,
  "Message": "اطلاعات وارد شده صحیح نیست"
}
```

مرحله دوم : انتقال کاربر به درگاه پرداخت

در صورتی که در مرحله اول، اطلاعات ارسالی صحیح بوده و توکن پرداخت را دریافت کرده باشید باید کاربر را به شیوه زیر به درگاه پرداخت انتقال دهید. برای این کار باید توکن دریافت شده (TOKEN) را در آخر آدرس زیر قرار داده و کاربر را به URL ایجاد شده ریدایرکت کنید تا بعد از چند ثانیه صفحه درگاه پرداخت را مشاهده کنید.

method: GET

URL: <https://my.raypay.ir/IPG?token=token>

بعد از این مرحله منتظر بمانید تا کاربر از صفحه پرداخت برگردد و مرحله سوم را در صفحه بازگشت خود پیاده سازی کنید.

مرحله سوم: بازگشت به سایت پذیرنده

بعد از انجام تراکنش، پرداخت کننده به آدرسی که پذیرنده در مرحله اول با پارامتر **redirectUrl** مشخص کرده بود منتقل می شود.

همچنین اطلاعات تفصیلی تراکنش به صورت **POST** به آدرس تعریف شده ارسال می شوند.

بعد از دریافت اطلاعات در سایت پذیرنده، حتما باید مرحله تایید تراکنش یا **verify** انجام شود تا پرداخت به صورت سیستمی در رای پی تکمیل شده و از بازگشت پول به پرداخت کننده جلوگیری شود.

به این منظور، تمامی پارامترهای دریافت شده در صفحه بازگشت، باید به صورت **Json** به آدرس زیر ارسال شوند:

method: POST

URL: <https://api.raypay.ir/raypay/api/v1/Payment/Verify>

پارامترهای ورودی

پارامتر	نوع	توضیحات
respcode	string	کد نتیجه انجام تراکنش
respmsg	string	متن نتیجه انجام تراکنش
amount	string	مبلغ کسر شده از حساب مشتری به ریال
invoiceid	string	شناسه پرداخت یا شناسه ارجاع بانکی رای پی
payload	string	اطلاعات تکمیلی ارسال شده توسط پذیرنده
terminalid	string	شماره ترمینال
tracenum	string	شماره پیگیری که بعد از تراکنش موفق به پذیرنده ارسال می‌گردد
hash	string	شماره کارت پرداخت کننده به صورت هش
datePaid	string	زمان و تاریخ انجام تراکنش
digitalreceipt	string	رسید دیجیتال
issuerbank	string	نام بانک صادر کننده ی دارنده کارت
cardnumber	string	شماره کارت پرداخت کننده به صورت Mask شده

خروجی متد

در صورتی که درخواست موفق باشد، وضعیت پاسخ با کد ۲۰۰ اعلام خواهد شد و پارامتر **data** در پاسخ ارسال می شود که شامل وضعیت تراکنش به همراه اطلاعات تراکنش می باشد.

```
{
  "statusCode": "Success",
  "isSuccess": true,
  "message": "string",
  "data": {
    "invoiceID": "string",
    "writtheadID": 0,
    "status": 0,
    "datePaid": "string",
    "comment": "string",
    "email": "string",
    "mobile": "string",
    "fullName": "string",
    "amount": 0,
    "factorNumber": "string"
  },
  "validationErrors": [
    {
      "propertyName": "string",
      "errors": [
        "string"
      ]
    }
  ]
}
```

در صورتی که سرویس در دسترس نباشد یا شناسه پرداخت وارد شده صحیح نباشد، وضعیت پاسخ، با کد ۴۰۰ اعلام خواهد شد و پارامتر **data** به صورت **Null** برمیگردد.

پارامترهای خروجی

پارامتر	نوع	توضیحات
status	int	وضعیت تراکنش. در صورتی که status برابر ۱ باشد به معنای پرداخت موفق و در غیر این صورت به معنای پرداخت ناموفق می باشد
invoiceID	string	شناسه پرداخت یا شناسه اجاع بانکی رای پی

شماره پیگیری رای پی	int	writheaderID
زمان و تاریخ انجام تراکنش	string	datePaid
توضیحات تراکنش	string	comment
ایمیل پرداخت کننده	string	email
شماره همراه پرداخت کننده	string	mobile
نام پرداخت کننده	string	fullName
مبلغ تراکنش	int	amount
شماره فاکتور یا شماره سفارش	string	factorNumber

نمونه کد عملیات بازگشت در سایت پذیرنده به زبان C# در زیر آمده است:

[HttpPost]

```
public async Task<IActionResult> Index(PaymentCallbackParam param)
{
    PaymentResultModel model = new PaymentResultModel();
    try
    {
        if (param != null)
        {
            var client = new HttpClient();
            var json = JsonConvert.SerializeObject(param);
            var requestContent = new StringContent(json, Encoding.UTF8, "application/json");
            try
            {
                HttpResponseMessage response = await client.PostAsync(
                    "https://api.raypay.ir/raypay/api/v1/Payment/verify", requestContent);
                HttpContent responseContent = response.Content;
                string result = await responseContent.ReadAsStringAsync();
            }
            catch (Exception exception)
            {
                throw exception;
            }
            int paymentStatus = Convert.ToInt32(JObject.Parse(result)["Data"]["Status"]);
            if (paymentStatus == 1)
                model.IsSuccess = true; // پرداخت موفق
            else
                model.IsSuccess = false; // پرداخت ناموفق
        }
    }
}
```

```
        model.FactorNumber =
JObject.Parse(result)["Data"]["FactorNumber"].ToString();
        model.InvoiceID = JObject.Parse(result)["Data"]["InvoiceID"].ToString();
        model.WritHeaderID =
JObject.Parse(result)["Data"]["WritheaderID"].ToString();
        model.Comments = JObject.Parse(result)["Data"]["comment"].ToString();
        model.Mobile = JObject.Parse(result)["Data"]["mobile"].ToString();
        model.Email = JObject.Parse(result)["Data"]["email"].ToString();
        model.FullName = JObject.Parse(result)["Data"]["fullName"].ToString();
        model.Message = JObject.Parse(result)["Message"].ToString();
        model.Amount = JObject.Parse(result)["Data"]["Amount"].ToString();
    }
    else
    {
        ModelState.AddModelError(string.Empty, "خطا در بازگشت از درگاه رخ داده است");
        return View();
    }
    catch (Exception ex)
    {
        ModelState.AddModelError(string.Empty, "خطا رخ داده است");
        return View(model);
    }
    return View(model);
}
```