



راهنمای نحوه اتصال به درگاه پرداخت اینترنتی رای پی

تیر ماه 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 پذیرنده صادر می شود که از بخش تنظیمات درگاه در پنل رای پی قابل دسترسی است:

- شناسه کاربری (User ID)
- کد پذیرنده (Acceptor Code)

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 و شماره ترمینال
- **مرحله دوم:** انتقال کاربر به صفحه پرداخت با توکن و شماره ترمینالی 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/getPaymentTokenWithUserID>

پارامترهای ورودی

| پارامتر | نوع | وضعیت | توضیحات |
|--------------|--------|---------|--|
| amount | String | ضروری | مبلغ تراکنش |
| 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 تنظیمات درگاه در پنل رای پی قابل دریافت می باشد. |
| acceptorCode | String | ضروری | کد پذیرنده که در بخش تنظیمات درگاه 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 |

| | | | |
|----------------------------|---------|--------|----------|
| نام پرداخت کننده | اختیاری | String | fullName |
| شماره همراه پرداخت کننده | اختیاری | String | mobile |
| پست الکترونیک پرداخت کننده | اختیاری | String | email |
| توضیحات تراکنش | اختیاری | String | comment |

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 **accesstoken** (توکن پرداخت) و **terminalID** (شماره ترمینال) می باشد.

```
{
  "statusCode": "Success",
  "isSuccess": true,
  "message": "string",
  "data": {
    "accesstoken": "string",
    "terminalID": "string"
  },
  "validationErrors": [
    {
      "propertyName": "string",
      "errors": [
        "string"
      ]
    }
  ]
}
```

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

مرحله دوم : انتقال کاربر به درگاه پرداخت

در صورتی که در مرحله اول، اطلاعات ارسالی صحیح بوده و دو مقدار **terminalID** و **accessToken** را دریافت کرده باشید باید کاربر را به درگاه پرداخت انتقال دهید. برای این کار ، باید نمونه فرم زیر را ایجاد کرده، مقادیر معتبر توکن و شماره ترمینال که در مرحله اول دریافت نمودید را در آن جایگذاری کرده و فرم را **submit** کنید.

```
<form method="post" action="https://mabna.shaparak.ir:8080/Pay">
<div>
<input type="text" name="TerminalID" value="22003751" />
<input type="text" name="token" value="fW6YDhWdKkpE6CxjHWYEq4N=" />
<input type="submit" value=" پرداخت " class="submit" />
</div>
</form>
```

برای راحتی کار، می توانید از تابع زیر که به زبان C# نوشته شده برای انجام عملیات ارجاع به درگاه پرداخت استفاده کنید.

```
public void postData(string accessToken, string terminalID)
{
    string FormName = "raypaySubmitForm";
    string Method = "post";
    string shaparakUrl = " https://mabna.shaparak.ir:8080/Pay ";
    var sb = new StringBuilder();
    sb.Append("<html><head>");
    sb.Append($"</head><body onload=\"document.{FormName}.submit()\">");
    sb.Append($"<form name=\"{FormName}\" method=\"{Method}\"");
    sb.Append($"action=\"{ shaparakUrl }\" >");
}
```

```
sb.Append($"<input name=\"TerminalID\" type=\"hidden\"  
value=\"{WebUtility.HtmlEncode(terminalID)}\">");  
sb.Append($"<input name=\"token\" type=\"hidden\"  
value=\"{WebUtility.HtmlEncode(accessToken)}\">");  
sb.Append("</form>");  
sb.Append("</body></html>");  
var data = Encoding.UTF8.GetBytes(sb.ToString());  
//modify the response  
var httpContext = _httpContextAccessor.HttpContext;  
var response = httpContext.Response;  
//change headers before the content is written to body  
response.OnStarting() =>  
{  
    response.ContentType = "text/html; charset=utf-8";  
    response.ContentLength = data.Length;  
    return Task.CompletedTask;  
});  
response.Clear();  
response.Body  
    .WriteAsync(data, 0, data.Length)  
    .Wait();  
}
```

بعد از این مرحله منتظر بمانید که 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، پارامتر **invoiceID** یا شناسه پرداخت، به صورت **query string** به انتهای آدرس بازگشت

اعلام شده توسط پذیرنده در مرحله اول اضافه می شود. مطابق زیر:

`https://Your-Redirect-URL?invoiceID={invoice_id}`

پس از دریافت پارامتر فوق (شناسه پرداخت)، جهت استعلام وضعیت تراکنش، باید متد زیر را فراخوانی کنید:

method: POST

URL: `https://api.raypay.ir/raypay/api/v1/Payment/checkInvoice?pInvoiceID={invoice_id}`

نکته: این متد، پارامتر شناسه پرداخت را به صورت **Query Parameter** دریافت می کند اما به علت

نوع متد (**POST**)، لازم است دیتای دلخواهی به صورت **Json** به همراه آن ارسال شود.

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": {
    "state": 0,
    "amount": 0,
    "invoiceID": "string",
    "factorNumber": "string",
    "mobile": "string",
    "comments": "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 |
|--------------|---------|---|
| state | binary | وضعیت تراکنش. در صورتی که state برابر ۱ باشد به معنای پرداخت موفق و در صورتی که برابر ۰ باشد به معنای پرداخت ناموفق می باشد |
| amount | integer | مبلغ تراکنش |
| invoiceID | String | شناسه پرداخت |
| factorNumber | String | شماره فاکتور یا شماره سفارش |
| mobile | String | شماره همراه پرداخت کننده |
| comments | String | توضیحات تراکنش |

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

```
public async Task<IActionResult> IndexAsync()
{
    PaymentResultModel model = new PaymentResultModel();
    try {
        if (!_httpContextAccessor.HttpContext.Request.Query["invoiceID"] != string.Empty)
        {
            string invoiceID = _httpContextAccessor.HttpContext.Request.Query["invoiceID"];
            var client = new HttpClient();
            var requestContent = new StringContent("data", Encoding.UTF8, "application/json");
            HttpResponseMessage response = await client.PostAsync(
                "https://api.raypay.ir/raypay/api/v1/Payment/checkInvoice?pInvoiceID=" +
```



```
invoiceID, requestContent);
HttpContent responseContent = response.Content;
var result = await responseContent.ReadAsStringAsync();

int paymentStatus = Convert.ToInt32(JavaScript.Parse(result)["Data"]["State"]);
if (paymentStatus == 1)
    model.IsSuccess = true; // پرداخت موفق
else
    model.IsSuccess = false ; // پرداخت ناموفق

model.FactorNumber =
JavaScript.Parse(result)["Data"]["FactorNumber"].ToString();
model.InvoiceID = JavaScript.Parse(result)["Data"]["InvoiceID"].ToString();
model.Comments = JavaScript.Parse(result)["Data"]["Comments"].ToString();
model.Mobile = JavaScript.Parse(result)["Data"]["Mobile"].ToString();
model.Amount = JavaScript.Parse(result)["Data"]["Amount"].ToString();
}
else
    Response.Redirect("~/", false);
}

catch (Exception ex)
{
    ModelState.AddModelError(string.Empty, "خطا رخ داده است",
    return View(model);
}

return View(model);
}
```