

وب سرویس هلو

معرفی محصول و راهنمای استفاده

سند فنی شماره 1

محاسب پرداز حکمت

نماینده چند منظوره طرفه نگار (هلو)

2022-1401

ver:1.7

api-1401-v1.7.odt

فهرست عناوین

- 3..... مقدمه
- 4..... وب سرویس هلو چیست؟
- 5..... بخش‌های مختلف وب سرویس
- 6..... راهنمای استفاده از **api**
- 7..... ایجاد طرف حساب جدید
- 8..... دریافت اطلاعات مشتریان (طرف حساب)
- 9..... دریافت اطلاعات مشتری (طرف حساب) بر اساس ایمیل
- 10..... دریافت اطلاعات کلیه محصولات
- 11..... دریافت اطلاعات محصولات آپدیت شده یک‌هفته اخیر
- 12..... دریافت تعداد محصول موجود
- 13..... دریافت تعداد آپدیت شده یک‌هفته اخیر
- 14..... دریافت تعداد محصولاتی که اصلاً آپدیت نشده اند
- 15..... ایجاد سفارش فروش در هلو
- 17..... دریافت اطلاعات محصول بر اساس کد محصول
- 18..... دریافت کلیه فاکتورها
- 19..... دریافت تعداد فاکتورها
- 20..... دریافت اطلاعات فاکتور بر اساس کد مشتری و تایپ فاکتور
- 21..... دریافت اطلاعات معین بر اساس کد کل و کد معین

مقدمه

این راهنما مخصوص استفاده کنندگان وب سرویس هلو نگاشته شده است. طبق این راهنما شما با نحوه نصب و راه اندازی بخش های مختلف آن آشنا خواهید شد.

توسعه دهندگان در استفاده از api ها چنانچه مشکلی و یا پیشنهادی دارند می توانند با شرکت محاسب پرداز حکمت و یا با ایمیل faalkhah@mohasebpardaz.com تماس حاصل نمایند.

پیشاپیش از حسن توجه شما سپاسگزاریم.

وب سرویس هلو چیست؟

تمامی استفاده کنندگان از نرم افزار حسابداری هلو می دانند که نرم افزار هلو بصورت محلی روی کامپیوترشان و تحت سیستم عامل ویندوز نصب می شود. البته بعضی از مشتریان محاسب پرداز ترجیح می دهند که از سیستم هاستینگ محاسب پرداز به جای نصب محلی استفاده کنند. با استفاده از این روش آن ها از هر جایی به نرم افزار خود دسترسی دارند. نرم افزار هلو فاقد هرگونه رابط ارتباطی (Interface) با دنیای بیرونی است. بنابراین شما به اطلاعات آن بیرون از خود نرم افزار دسترسی ندارید.

در حال حاضر بسیاری از خریداران نرم افزار هلو از وب سایت های فروشگاه های مخصوص به خود استفاده می کنند. بدیهی است که اتصال فروشگاه به نرم افزار هلو یک نیاز ضروری بود.

از آنجائیکه اکثر فروشگاهها برای راه اندازی فروشگاه خود از سیستم مدیریت محتوی (CMS) وردپرس به همراه فروشگاه ساز ووکامرس بهره می برند، در این مقطع نیاز به ساخت افزونه وردپرس بیشتر احساس می شد، از اینرو در حال حاضر اتصال وردپرس به نرم افزار هلو بطور کامل پیاده سازی شده و مورد استفاده بسیاری از شرکت ها قرار گرفته است. اما باید یادآور شد که طراحی کلی این وب سرویس بگونه ای انجام شده است تا بتوان در سایر وب سایت فروشگاه های نیز از آن بهره برد.

بخش های مختلف وب سرویس

این وب سرویس از دو بخش اصلی زیر تشکیل شده است.

- **هلو وب (سرویس دهنده):** هلو وب نرم افزار رابط مابین هلو و فروشگاههاست. این نرم افزار با استفاده از تکنولوژی api به تمامی درخواست های مورد نیاز جواب می دهد و اطلاعات لازم را در اختیار می گذارد. این نرم افزار بر روی کامپیوتری که روی آن ویندوز نصب شده است کار می کند. این سرویس به دو شکل می تواند روی سیستم عامل ویندوز مستقر گردد. یا آن را می توان روی IIS نصب کرد یا می توان بصورت سرویس ویندوز و بر اساس پورت خاصی راه اندازی نمود. در عین حال هلو وب می تواند بر روی سایر سیستم عامل ها از جمله لینکس نیز نصب شود، اما از آنجائیکه هلو برای ویندوز طراحی شده است و بانک اطلاعاتی آن نیز روی ویندوز هست لذا پیشنهاد می شود هلو وب بر روی ویندوز نصب شود.
- **افزونه ها (سرویس گیرنده):** از آنجائیکه هلو وب از تکنولوژی API برخوردار هست لذا می تواند به تمامی تکنولوژی های جانبی از قبیل افزونه های فروشگاهها که مثلاً روی ورد پرس نصب و راه اندازی می شود جوابگو باشد. از طرفی می تواند جوابگوی سایر نیازها از قبیل باشگاه مشتریان، CRM و غیره باشد. در حال حاضر افزونه مخصوص وب سایت فروشگاههای وردپرس بر اساس و کامرس طراحی و ساخته شده است. بنابر این مشتری پس از خرید وب سرویس، می تواند افزونه مخصوص به خود را دریافت کرده و بر روی فروشگاه خود نصب نماید. هر چند بسیاری از مشتریان ترجیح می دهند تا این افزونه توسط شرکت محاسب پرداز نصب و راه اندازی شود.

راهنمای استفاده از api

پس از خرید وب سرویس، تعداد یک عدد توکن نرم‌افزاری به شما ارائه می‌شود تا در توسعه سیستم‌های جانبی از قبیل پلاگین‌ها، باشگاه مشتریان و غیره مورد استفاده قرار گیرد. مشتریانی که پلاگین ورد پرس را خریداری می‌نمایند نیازی به این توکن ندارند و در واقع توکن در درون پلاگین قرار خواهد گرفت.

مشتریان عزیزی که قصد توسعه سرویس‌های جانبی را دارند توجه داشته باشند که می‌توانند برای تست‌های خود از نرم‌افزار postman و یا مشابه‌های آن برای تست خود استفاده نمایند. نکته فنی دیگر اینکه در درون کدهای برنامه نویسی برای ارسال توکن به وب سرویس توجه داشته باشید که مابین کلمه Bearer تعداد یک عدد اسپیس با توکن قرار گیرد.

ایجاد طرف حساب جدید

Desc	ایجاد طرف حساب در هلو(مشتری) توضیح اینکه توسعه دهندگانی که مایل هستند از سرویس api برای فروشگاهشان استفاده کنند ممکن است مجبور شوند که ابتدای فیلدهای بخش پارامتر را با کاراکترهای بزرگ ارسال کنند مثلاً بجای cName از Cname استفاده کنند.
Method	Post
Url	http://holoop/api/customer/create example: http://localhost:5000/api/customer/create
Params	Authorization: `Bearer ` + `token` { "cname": string, "email": string, "cbillingaddress": string, "cbillingtel": string, "col_code_bed": int, "col_code_bes": int, "city_code": int }
Result	Status code 200
Erorr	1

دریافت اطلاعات مشتریان (طرف حساب)

Desc	دریافت اطلاعات همه مشتریان
Method	Get
Url	http://holooip/api/customer/getall example: http://localhost:5000/api/customer/getall
Params	Authorization: `Bearer ` + `token`
Result	[{ "c_Code": "00001", "c_Name": "ترخیص کار", "c_Code_C": "00001", "c_AliasName": "", "c_Address": "", "c_Tel": "", "c_Fax": "", "c_Mobile": "", "economic_Code": "", },]
Erorr	1

دریافت اطلاعات مشتری (طرف حساب) بر اساس ایمیل

Desc	دریافت اطلاعات مشتری
Method	Get
Url	http://holooip/api/customer/getbymobile?mobile=xxx example: http://localhost:5000/api/customer/getbymobile?mobile=091xxxxxxxx
Params	Authorization: `Bearer ` + `token`
Result	{ "c_Code": "00027", "c_Name": "آقای داود مستعد", ... }
Erorr	1,

دریافت اطلاعات کلیه محصولات

Desc	دریافت اطلاعات کلیه محصولات موجود در هلو.
Method	Get
Url	http://holooip/api/article/getall example: http://localhost:5000/api/article/getall
Params	Authorization: `Bearer ` + `token`
Result	[<pre> { "a_Code": "0101001", "a_Code_C": "10216a", "a_Name": "product-one", "exist": 1500, "sel_Price": 90000, "m_groupname": "m-groupname", "s_groupname": "s-groupname" }, { ... } </pre>]
Erorr	1

دریافت اطلاعات محصولات آپدیت شده یکپهفته اخیر

Desc	<p>دریافت اطلاعات محصولاتی از هلو که ظرف یکپهفته گذشته آپدیت شده اند علت اینکه یکپهفته در نظر گرفته شده است بدلیل احتمال در قطعی سرویس های ارتباطی از قبیل اینترنت؛ مشکلات عدم ارتباط هلووب با فروشگاه یا بالا نبودن سرویس فروشگاه و غیره می باشد. بنابر این در حال حاضر تنها آپدیت های یکپهفته گذشته ملاک عمل هست. ممکن است در آینده امکانات بیشتری به این قسمت از قبیل اینکه کاربر خود بتواند مدت زمان را از طریق فروشگاه تعیین کند بوجود بیاید. ولی در این ورژن فعلاً همین یکپهفته گذشته مد نظر می باشد.</p>
Method	Get
Url	<p>http://holoop/api/article/getalldited example: http://localhost:5000/api/article/getalldited</p>
Params	Authorization: `Bearer ` + `token`
Result	<pre>[{ "a_Code": "0101001", "a_Code_C": "10216a", "a_Name": "product-one", "exist": 1500, "sel_Price": 90000, "m_groupname": "m-groupname", "s_groupname": "s-groupname" },]</pre>
Errorr	1

دریافت تعداد محصول موجود

Desc	دریافت تعداد محصولات موجود در هلو
Method	Get
Url	http://holooip/api/article/getallcount example: http://localhost:5000/api/article/getallcount
Params	Authorization: `Bearer ` + `token`
Result	nnn
Erorr	1

دریافت تعداد آپدیت شده یک هفته اخیر

Desc	دریافت تعداد محصولات آپدیت شده یک هفته اخیر موجود در هلو
Method	Get
Url	http://holooip/api/article/getallcount example: http://localhost:5000/api/article/getallcount
Params	Authorization: `Bearer ` + `token`
Result	nnn
Erorr	1

دریافت تعداد محصولات که اصلاً آپدیت نشده اند

Desc	دریافت تعداد محصولات که اصلاً آپدیت نشده اند
Method	Get
Url	http://holooip/api/article/getallnullcount example: http://localhost:5000/api/article/getallnullcount
Params	Authorization: `Bearer ` + `api01:token`
Result	nnn
Erorr	1

ایجاد سفارش فروش در هلو

Desc	<p>ایجاد سفارش فروش در هلو (سفارش های ایجاد شده در فروشگاه به هلو ارسال می شوند)</p> <p>توضیح اینکه توسعه دهندگانی که مایل هستند از سرویس api برای فروشگاهشان استفاده کنند ممکن است مجبور شوند که ابتدای فیلدهای بخش پارامتر را با کاراکترهای بزرگ ارسال کنند مثلاً بجای cName از CName استفاده کنند.</p>
Method	Post
Url	<p>http://holoop/api/RQ/create</p> <p>example: http://localhost:5000/api/RQ/create</p>
Params	<pre> Authorization: `Bearer ` + `token` { "RQTitle" : { "cName" : "شهرام میردامادیان - shahram.mirdamadian@gmail.com", "email" : "shahram.mirdamadian@gmail.com", "sumPrice" : "1000", "shop_user_name" : "shop", "paid" : "yes", "payment_method" : "cod" }, "RQDetail" : [{ "r_ArCode" : "0102002", "r_Few" : "1", "r_Commen" : "" }] } </pre>

Result	Number of inserted
Erorr	1

دریافت اطلاعات محصول بر اساس کد محصول

Desc	دریافت اطلاعات محصول از قبیل موجود و غیره بر اساس کد محصول از هلو
Method	Post
Url	http://holoop/api/article/getbycode?code=xxxxxx example: http://localhost:5000/api/article/getbycode?code=0101001
Params	Authorization: `Bearer ` + `token`
Result	{ "a_Code": "0101001", "a_Code_C": "10216a", "a_Name": "لوله آب 16 (AA).", "a_Country": "", ... }
Erorr	1

Desc	دریافت کلیه فاکتورها از هلو
Method	Get
Url	http://holoop/api/facture/getall example: http://localhost:5000/api/facture/getall
Params	Authorization: `Bearer ` + `token`
Result	{ "fac_Code": "000004", "fac_Type": "F", "fac_Code_C": "4", "c_Code": "00031", ... }
Erorr	1

دریافت تعداد فاکتورها

Desc	دریافت تعداد فاکتورها از هلو
Method	Get
Url	http://holooip/api/facture/getallcount example: http://localhost:5000/api/facture/getallcount
Params	Authorization: `Bearer ` + `token`
Result	100
Erorr	1

دریافت اطلاعات فاکتور بر اساس کد مشتری و تایپ فاکتور

Desc	دریافت کلیه فاکتورها از هلو بر اساس کد مشتری و تایپ فاکتور. تایپ فاکتور می‌تواند K به معنای خرید و F به معنای فروش باشد. و یا می‌توان نال ارسال کرد که در این صورت کلیه فاکتورهای خرید و فروش را لیست خواهد کرد
Method	Get
Url	http://holoop/api/facture/getbycustomercode? c_code=xxx&fac_type=x example: http://localhost:5000/api/facture/getbycustomercode? c_code=00017&fac_type=F
Params	Authorization: `Bearer ` + `token`
Result	{ "fac_Code": "000004", "fac_Type": "F", "fac_Code_C": "4", "c_Code": "00017", ... }
Erorr	1

دریافت اطلاعات معین بر اساس کد کل و کد معین

Desc	دریافت اطلاعات معین بر اساس کد کل و کد معین
Method	Get
Url	http://holooip/api/rpt/getmoien?col_code=x&moien_code=x example: http://localhost:5000/api/rpt/getmoien?col_code=103&moien_code=0019
Params	Authorization: `Bearer ` + `token`
Result	<pre>{ "detail" : [{ ... "bed" : 6061999955, ... },], "total" : { ... "besTotal" : 0, ... } }</pre>
Erorr	1