

به نام خدا

راهنمای استفاده از وب سرویس های فروشگاه آنلاین بازارا

نکات عمومی در استفاده از سرویس ها :

کلید وب سرویس ها به صورت Rest پیاده سازی شده اند و فراخوانی آنها صرفا شامل دستورهای Get و Post است. Content-type همه سرویس ها بجز سرویس آپلود برای request و response از نوع application/json است.

سرویس آپلود جداگانه توضیح داده خواهد شد.

تمام سرویس ها بجز سرویس Login نیاز به پارامتر userToken دارند که برای کنترل دسترسی و شناسایی کاربر استفاده می شود.

برای دریافت userToken از سرویس Login استفاده می شود که در پادامه توضیح داده می شود.

در تمام سرویس های ارسال و دریافت اطلاعات پارامتر systemSyncID نیز استفاده می شود که شماره شناسایی یک پایگاه داده به صورت منحصر به فرد است.

مقدار این پارامتر برای هر کابر در حال حاضر توسط شرکت محک ساخته و در اختیار مشتریان قرار می گیرد.

در تمام سرویس های دریافت اطلاعات پارامتر changedAfter نیز استفاده می شود که برای دریافت داده هایی که از آخرین بار دریافت تغییر کرده اند کاربرد دارد. نحوه استفاده از این پارامتر در ادامه توضیح داده می شود.

اغلب پارامترهای مورد نیاز برای فراخوانی سرویس ها منجمله سه مورد توضیح داده شده در بالا می توانند از طریق query string در url فراخوانی سرویس نیز فرستاده شوند. مثلا اگر سرویسی با استفاده از Http Get فراخوانی شود مقادیر پارامترهای ذکر شده در بالا باید از طریق query string ارسال شوند.

آدرس پایه فراخوانی وب سرویس ها <http://bazaraservices.mahaksoft.com/sync> است.

نمونه فراخوانی و استفاده از سرویس ها به زبان C# به همراه این فایل ارائه می شود.

سرویس Login:

همه سرویس ها برای فراخوانی نیاز به پارامتر userToken دارند که برای بدست آوردن آن باید سرویس Login را فراخوانی کرد.

http://bazaraservices.mahaksoft.com/sync/login	url سرویس
Get, Post	نوع متد
<pre>{ "username": "farzadf", "password": "846328" }</pre>	نمونه محتوای درخواست post
<pre>{ "result": "success", "code": 0, "message": "", "data": { "token": "R9qQvB5_f9izMyVj68XgQosNTe-3ugiG4nOsMS" } }</pre>	نمونه محتوای پاسخ دریافتی

نمونه کد فراخوانی وب سرویس در C#:

```

private string CallPostWebService(string, string jsonContent)
{
    using (var client = new HttpClient())
    {
        client.BaseAddress = new Uri("http://bazaraservices.mahaksoft.com");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(
            new MediaTypeWithQualityHeaderValue("application/json"));

        var content = new StringContent(jsonContent, Encoding.UTF8,
            "application/json");

        var response = client.PostAsync(serviceUrl, content).Result;
        if (response.IsSuccessStatusCode)
            return response.Content.ReadAsStringAsync().Result.ToString();

        return "error";
    }
}

```

```

private string CallGetWebService(string serviceUrl)
{
    using (var client = new HttpClient())
    {
        client.BaseAddress = new Uri("http://bazaraservices.mahaksoft.com");
        client.DefaultRequestHeaders.Accept.Clear();
        client.DefaultRequestHeaders.Accept.Add(
            new MediaTypeWithQualityHeaderValue("application/json"));

        var response = client.GetAsync(serviceUrl).Result;
        if (response.IsSuccessStatusCode)
            return response.Content.ReadAsStringAsync().Result.ToString();

        return "error";
    }
}

```

نمونه فراخوانی سرویس Login با استفاده از متدهای ساخته شده در بالا :

اجرا از طریق متد GET و ارسال پارامترهای مورد نیاز از طریق query string

```
string serviceResult = CallGetWebService("\\Sync\\Login?username=" + txtUsername.Text + "&password=" + txtPassword.Text);
```

بعد از اجرای کد فوق متغیر serviceResult حاوی رشته json نتیجه خواهد بود که می توانید به طرق مختلف آن را parse کرده و مقادیر مورد نظر خود را استخراج کنید. به عنوان مثال نحوه خواندن مقدار userToken از رشته فوق با استفاده از کتابخانه Newtonsoft.Json و به صورت dynamic در زیر آمده است.

```
dynamic obj = JObject.Parse(serviceResult);  
string userToken = obj.data.token;
```

اجرا از طرق متد POST

برای اجرای از طریق متد POST باید ابتدا رشته json محتوای درخواست را بسازیم که این هم از طرق مختلف امکان پذیر است. به عنوان مثال دو روش ساخت رشته به صورت دستی و ساخت از طریق serialize کردن در زیر آمده است.

```
string requestJson = "{\"username\": \"" + txtUsername.Text + "\", \"password\": \"" + txtPassword.Text + "\"}";
```

```
string requestJson = JsonConvert.SerializeObject(new { username = txtUsername.Text, password = txtPassword.Text });
```

حالا می توانید سرویس Login را از طریق متد POST به شکل زیر اجرا کنید

```
string serviceResult = CallPostWebService("\\Sync\\Login", requestJson);
```

نکته : توکن بدست آمده از طریق سرویس لاگین فقط برای مدت محدودی اعتبار دارد و بعد از انقضا باید توکن جدیدی با فراخوانی مجدد سرویس لاگین بدست آورد. این مدت در حال حاضر ۳۰ دقیقه است که ممکن است در آینده تغییر کند. بهتر است قبل از هر مجموعه عملیات برای بروزرسانی اطلاعات سرویس لاگین فراخوانی شود تا در حین بروزرسانی با خطای توکن منقضی شده مواجه نشوید.

نکته : توکن ها همچنین محدود به IP هستند و پس از بدست آوردن توکن جدید فراخوانی سرویس های بعدی با آن توکن فقط از طریق همان IP که توکن با آن بدست آمده امکانپذیر است.

ارسال و دریافت اطلاعات :

بعد از دریافت توکن می توانید با استفاده از سایر وب سرویس ها اطلاعات را از سرور بخوانید و یا در سرور ذخیره کنید. نکته مهمی که در پیاده سازی سیستم باید مد نظر قرار بدهید این است که فرایند سینک (به روز رسانی) داده ها در افزونه فروشگاه آنلاین بازارا در نرم افزار محک در اغلب جاها به صورت یک طرفه پیاده سازی شده است و هرچند شما می توانید از همه وب سرویس ها به صورت دوطرفه در سایت خود استفاده کنید ولی افزونه بازار ممکن است داده های شما را override کند.

در ادامه به قوانین استفاده شده در فرآیند بروزرسانی خواهیم پرداخت

ارسال اطلاعات کالاها به صورت یکطرفه از سمت نرم افزار حسابداری محک به سمت وب سایت است. در افزونه بازارا در نرم افزار محک کاربر کالاهایی که مد نظر دارد تا در وب سایت خود به فروش برسند را انتخاب می کند و افزونه اطلاعات آن کالاها را به سرور ارسال می کند و بعد از هر تغییر کالا در نرم افزار حسابداری داده های تغییر کرده را در سرور به روز رسانی میکند. بنابراین اگر شما در وب سایت خود کالای جدیدی تعریف کنید و با استفاده از وب سرویس های ارسال اطلاعات کالا، اطلاعات آن را به سرور ارسال کنید افزونه بازارا اطلاعات آن کالا را دریافت نخواهد کرد.

بروز رسانی اطلاعات اشخاص به صورت خاص در افزونه بازارا پیاده سازی شده است (شبه دو طرفه). در افزونه بازارا کاربر افرادی را که می خواهد به عنوان کاربر سایت نیز باشند و بتوانند به فروشگاه آنلاین لاگین و اقدام به خرید کنند را انتخاب می کند سپس افزونه اطلاعات این افراد را به سرور ارسال می کند و بعد از هر تغییر در اطلاعات آن اشخاص داده های جدید برای آنها ارسال خواهد شد. اما بر خلاف کالاها اگر در سایت نیز کاربر جدیدی ایجاد شود و اطلاعات آن به سرور ارسال شود افزونه بازارا اطلاعات آن شخص جدید را دریافت می کند و به کاربر محک نشان می دهد و کاربر محک می تواند در صورت تمایل آن شخص را به لیست اشخاص در نرم افزار محک اضافه نماید. در صورت تبدیل شخص ارسال شده از سایت به شخص در نرم افزار محک اطلاعات آن شخص نیز بعد از هر تغییر مجدداً توسط افزونه بازارا به سرور ارسال می شود.

اما نکته ای که باعث می شود سینک اطلاعات افراد به صورت شبه دوطرفه باشد این است که افزونه بازارا فقط اطلاعات افراد جدید را سرور دریافت می کند و اگر در سایت اطلاعات شخص تغییر کند و به سرور ارسال شود افزونه بازار تغییرات را دریافت نخواهد کرد و بعد از انجام عملیات روز رسانی اطلاعات در افزونه بازار اطلاعات قبلی شخص که در نرم افزار محک ثبت شده اند به سرور ارسال می شود.

ارسال اطلاعات فاکتورها و دریافتی ها به صورت یکطرفه از سمت وب سایت به سمت نرم افزار محک است. در وب سایت پس از آن که کاربری خرید انجام داد اطلاعات آن خرید از طریق وب سرویس های مربوطه باید به سرور ارسال شود. بعد از آن افزونه بازارا آن اطلاعات را دریافت کرده و به کاربر محک نمایش می دهد و کاربر محک در صورت تمایل می تواند خرید انجام شده را به صورت فاکتور در نرم افزار ثبت کند و یا آن خرید را حذف آن لغو نماید. نکته : ارسال اطلاعات فاکتور به صورت یکطرفه از سمت سایت به سمت نرم افزار است به استثنای وضعیت فاکتور که در صورت تایید فاکتور در سمت نرم افزار و یا لغو آن وضعیت سفارش مربوطه در سرور به روز رسانی می شود و در صورت نیاز وب سایت می تواند وضعیت جدید سفارش را خوانده و متناسب با آن عمل نماید.

ارسال و دریافت اطلاعات کالاها

برای ارسال دریافت اطلاعات کالاها از مجموعه سرویس های زیر استفاده می شود که در این مستند فقط سرویس های مربوط به دریافت اطلاعات کالا توضیح داده می شوند.

توضیحات	url سرویس
دریافت لیست کالاهای ثبت شده در سرور	/Sync/GetProducts
ارسال و ثبت اطلاعات کالا در سرور	/Sync/SaveProduct , /Sync/SaveProducts
حذف کالا از سرور	/Sync/DeleteProduct , /Sync/DeleteProducts
دریافت جزئیات قیمت و موجودی کالا	/Sync/GetPrices
ارسال و ثبت جزئیات قیمت و موجودی کالا در سرور	/Sync/SavePrice , /Sync/SavePrices
حذف جزئیات قیمت و موجودی کالا از سرور	/Sync/DeletePrice , /Sync/DeletePrices
دریافت لیست گروه های کالا از سرور	/Sync/GetCategories
ارسال و ثبت اطلاعات گروه کالا در سرور	/Sync/SaveCategory , /Sync/SaveCategories
حذف گروه کالا از سرور	/Sync/DeleteCategory , /Sync/DeleteCategories
حذف عکس کالا از سرور	/Sync/DeleteProductImage
دریافت لیست عکس های ثبت شده	/Sync/GetPictures

نکته : سرویس هایی که url آن به صورت جمع است برای ارسال و یا دریافت همزمان اطلاعات چندین شی به کار می رود. به عنوان مثال SaveProduct برای ارسال و ثبت اطلاعات یک کالا و SaveProducts برای ارسال اطلاعات چندین کالا به صورت همزمان استفاده می شود.

در ارسال و ثبت اطلاعات وابستگی اشیاء به هم و رابطه Master/Detail آنها باید در نظر گرفته شود. به عنوان مثال برای ارسال اطلاعات کالاها ابتدا باید اطلاعات گروه های کالا ثبت شود بعد از آن اطلاعات کالا و در نهایت اطلاعات جزئیات قیمت و موجودی کالا باید ارسال و ثبت شوند. در مورد حذف اطلاعات این روند برعکس است و ابتدا باید جزئیات حذف شوند سپس اطلاعات کالا و در نهایت گروه های کالا حذف شوند.

در ادامه سرویس هایی که برای دریافت اطلاعات کالاها و استفاده در سایت کاربرد دارند توضیح داده خواهند شد.

سرویس `/Sync/GetProducts`:

این سرویس برای دریافت اطلاعات عمومی کالاهای ثبت شده در سرور به کار می رود

http://bazaraservices.mahaksoft.com/sync/GetProducts	url سرویس
Get, Post	نوع متد
<pre>{ "systemSyncID": 1, "changedAfter": 100025, "userToken": "R9qQvB5_f9izMyVj68XgQosNTe" }</pre>	نمونه محتوای درخواست post
<pre>{ "result": "success", "data": [{ "ProductID": \, "Code": 10, "Name": "forbidden product", "UnitName": "عدد", "CreateDate": "2016-02-17T13:31:00.863", "UpdateDate": "2016-02-17T13:31:00.97", "Comment": "my comment", "Deleted": false, "RowVersion": 10013 }] }</pre>	نمونه محتوای پاسخ دریافتی

نکته: `ProductID` فیلد کلید کالا در سرور است. در هنگام استفاده از سرویس های ارسال داده به سرور در صورتی که فیلد کلید شی مربوطه مقدار داشته باشد شی آپدیت می شود و اگر مقدار صفر داشته باشد شی جدید ساخته می شود.

فیلد `Code` کد کالا در نرم افزار حسابداری محک است

سرویس :/Sync/GetPrices

این سرویس برای دریافت اطلاعات جزئیات کالاها شامل قیمت و موجودی با مشخصه های آن به کار می رود

http://bazaraservices.mahaksoft.com/sync/GetPrices	url سرویس
Get, Post	نوع متد
<pre>{ "systemSyncID": 1, "changedAfter": 100025, "userToken": "R9qQvB5_f9izMyVj68XgQosNTe" }</pre>	نمونه محتوای درخواست post
<pre>{ "result": "success", "data": [{ "ProductPriceID": 1, "ProductID": 1, "Properties": "base64 encoded json string", "AvailableCount": 12, "DefaultPriceLevel": 1, "Price1": 10000, "Price2": 12000, "Price3": 12000, "Price4": 12000, "Price5": 12000, "Price6": 12000, "Price7": 12000, "Price8": 12000, "Price9": 12000, "Price10": 12000, "CreateDate": "2016-02-17T13:54:15.273", "UpdateDate": "2016-02-17T13:54:15.283", "Deleted": false, "RowVersion": 10016 }] }</pre>	نمونه محتوای پاسخ دریافتی

نکته Price1 تا Price10 سطوح قیمتی کالا در نرم افزار حسابداری هستند و در صورتی که بخواهید برای مشتریان

مختلف از قیمت های متفاوت استفاده کنید کاربرد دارد. در حال حاضر برای هر ProductID یک ProductPriceID وجود

دارد ولی در آینده با تکمیل قسمت جزئیات کالا در نرم افزار محک این رابطه یک به چند خواهد بود. مثلا کالای گوشی

آیفون با مشخصه رنگ زرد ۱۰ عدد موجود است با قیمت های ... ریال و از همان کالا با مشخصه رنگ آبی با گارانتی یک ساله

۵ عدد موجود است با قیمت های ...

سرویس :/Sync/GetPictures

این سرویس برای دریافت اطلاعات عکس های کالاهای ثبت شده در سرور به کار می رود

http://bazaraservices.mahaksoft.com/sync/GetPictures	url سرویس
Get, Post	نوع متد
<pre>{ "systemSyncID": 1, "changedAfter": 100025, "userToken": "R9qQvB5_f9izMyVj68XgQosNTe" }</pre>	نمونه محتوای درخواست post
<pre>{ "result": "success", "data": { "PictureID": 1, "ItemID": 112, "ItemType": 1, "PictureCode": 12, "FileName": "", "DisplayOrder": 0, "Title": "", "Url": "", "FileSize": 1024, "Width": 100, "Height": 100, "Format": "jpg", "Sha1": "", "UpdateDate": "", "CreateDate": "", "Deleted": false, "RowVersion": 100 } }</pre>	نمونه محتوای پاسخ دریافتی

نکته : این سرویس به صورت کلی برای دریافت لیست عکس ها به کار می رود و مختص کالاها نیست. فیلد ItemType

مشخص کننده نوع شی است که عکس به آن تعلق دارد و ItemID فیلد کلید آن شی در سرور است.

مثلا در مورد کالاها ItemType برابر با ۱ است و ItemID برابر با ProductID کالای مربوطه است.

نکته : ممکن است فیلدهای Url و Sha1 خالی باشند. این به این معنا است که اطلاعات عکس ارسال شده ولی فایل محتوی

عکس هنوز آپلود نشده است

ارسال و دریافت اطلاعات اشخاص

سرویس : /Sync/GetPeople

این سرویس برای دریافت لیست اشخاص ثبت شده در سرور کاربرد دارد

url سرویس	http://bazaraservices.mahaksoft.com/sync/GetPeople
نوع متد	Get, Post
نمونه محتوای درخواست post	<pre>{ "systemSyncID": 1, "changedAfter": 100025, "userToken": "R9qQvB5_f9izMyVj68XgQosNTe" }</pre>
نمونه محتوای پاسخ دریافتی	<pre>{ "result": "success", "data": [{ "PersonID": 136, "PersonType": 0, "PersonCode": 2, "PersonGroupID": 108, "FirstName": "فروشگاه", "LastName": "سیتی سنتر مهزیار (کالا)", "Gender": 0, "NationalCode": null, "DefaultAddressID": null, "Mobile1": null, "Phone1": null, "Fax": null, "Email": "T1@test.Test", "UserName": "2", "Password": null, "PriceLevel": 0, "CityCode": -1, "Credit": -1, "Balance": 0, "CreateDate": "2016-06-02T14:16:23.5", "UpdateDate": "2016-06-02T14:16:23.507", "Comment": null, "Deleted": false, "RowVersion": 203790 }] }</pre>

PersonID فیلد کلید شخص در سرور و PersonCode کد شخص در محک است. در هنگام ثبت شخص جدید در سایت فیلد PersonCode را برابر با ۰ قرار دهید، در صورت تبدیل این شخص در نرم افزار محک به شخص محکی این فیلد در آنجا مقدار گرفته و در سرور آپدیت می شود.

PersonGroupID کد گروه شخص در سرور است و DefaultAddressID فیلد کلید آدرس پیش فرض برای شخص است. برای هر شخص بیشتر از یک آدرس می توان ثبت کرد که در ادامه سرویس مربوطه توضیح داده خواهد شد. مثلا در هنگام خرید کالا از سایت شخص می تواند یک آدرس جدید برای دریافت سفارش خود ثبت کند.

PriceLevel شماره سطح قیمتی است که برای این شخص استفاده می شود. اگر این مقدار برابر با ۰ باشد برای این شخص از سطح قیمت مشخص شده در گروه شخص استفاده می شود و اگر اگر آنجا هم ۰ بود از سطح قیمت پیش فرض مشخص شده در جزئیات کالا استفاده می شود.

مقدار CityCode همان مقادیر ثبت شده در جدول Zone در نرم افزار محک است.

فیلد Deleted که در همه اشیا وجود دارد نشان دهنده حذف شدن شی مربوطه است.

فیلد RowVersion که در همه اشیا وجود دارد در واقع مقدار عددی timestamp رکورد مربوطه در دیتابیس است و پارامتر changedAfter که در فراخوانی اغلب سرویس ها وجود دارد به همین فیلد اشاره می کند.

به عنوان مثال شما لیست همه افراد و یا هر شی دیگری را از سرور دریافت کرده اید، بزرگترین مقدار RowVersion را در جایی در سیستم لوکال خود ذخیره کنید، بعد از آن برای دریافت اطلاعات جدید و بروز رسانی وب سایت خود هنگام فراخوانی سرویس های دریافت داده مقدار ذخیره شده را در قالب پارامتر changedAfter به سرور ارسال کنید، حالا سرور فقط اطلاعاتی را به شما بر می گرداند که بعد از آخرین باری که داده ها را خوانده اید تغییر کرده باشند. این مورد تاثیر زیادی در پرفورمنس سرور و هم کلاینت استفاده کننده از سرویس ها خواهد داشت.

سرویس /Sync/SavePeople :

این سرویس برای ذخیره گروهی اطلاعات افراد در سرور استفاده می شود

url سرویس	http://bazaraservices.mahaksoft.com/sync/SavePeople
نوع متد	Post
نمونه محتوای درخواست post	<pre>{ "systemSyncID": 1, "useTransaction": true, "userToken": "R9qQvB5_f9izMyVj68XgQosNTe-3ugiG4n0sMS", "people": [{ "PersonID": 0, "FirstName": "Javad", "LastName": "Javadi", "PersonType": 0, "PersonCode": 100, "PersonGroupID": null, "Gender": 0, "NationalCode": "2232", "Mobile1": "", "Phone1": "", "Fax": "", "Email": "", "UserName": "", </pre>

<pre> "Password": "", "PriceLevel": 1, "CityCode": null, "Credit": 0, "Balance": 0, "Comment": "", "Address": { "AddressLabel": "this is my address", "CityCode": 102000, "Phone": "05132105521", "Longitude": 0, "Latitude": 0, "PostalCode": "1234567890", "Title": "Home Address", "Comment": "" } }] } </pre>	
<pre> { "result": "success", "data": { "successCount": 1, "results": [{ "Result": true, "Index": 1, "EntityID": 159 }] } } </pre>	<p>نمونه محتوای پاسخ دریافتی</p>
<pre> { "result": "error", "data": { "successCount": 0, "results": [{ "Result": false, "Index": 1, "Errors": [{ "Property": "PersonGroupID", "Error": "invalid PersonGroupID" }], "AdditionalData": "Optional data" }] } } </pre>	<p>نموی محتوای پاسخ دریافتی وقتی بخ خطا بر می خورد</p>

نکته فرمت خروجی اکثر سرویس های ثبت داده و حذف داده در سرور مشابه هم و مانند مثال فوق است.

Index شماره اندیس آیتمی است که نتیجه برای آن برگشته که به اندیس شی ارسال شده در آرایه اطلاعات اشیا اشاره دارد.

نکته : Index از یک شروع میشود

EntityID فیلد کلید شی در سرور است که برای فراخوانس سرویس های بعدی که فیلد کلید شی را لازم دارند کاربرد دارد

Error آرایه ای از لیست خطاهایی است که در هنگام ثبت اطلاعات به وقوع پیوسته

فرمت برگشتی مثال بالا برای حالت ثبت گروه اشیا است و برای ثبت تکی اشیا به شکل زیر خواهد بود.

Entity Save Result in Single Object Mode

```
{
  "Result": true,
  "EntityID": 12,
  "Errors": [
    "Property Error Format"
  ],
  "AdditionalData": "Optional with custom format"
}
```

Property Error

```
{
  "Property": "Title",
  "Code": 1,
  "Error": "Title is empty"
}
```

در صورتی که اطلاعات اضافی وجود داشته باشد از طریق فیلد AdditionalData مشخص خواهد شد.

فرمت AdditionalData بسته به سرویس مربوطه متغیر است

سرویس :/Sync/GetAddresses

این سرویس دریافت اطلاعات آدرس های اشخاص استفاده می شود

http://bazaraservices.mahaksoft.com/sync/GetAddresses	url سرویس
Get, Post	نوع متد
<pre>{ "systemSyncID": 1, "changedAfter": 100025, "userToken": "R9qQvB5_f9izMyVj68XgQosNTe" }</pre>	نمونه محتوای درخواست post
<pre>{ "result": "success", "code": 0, "message": "", "data": [{ "AddressID": 1, "AddressCode": null, "PersonID": 2, "CityCode": 102000, "PostalCode": "1234567890", "AddressLabel": "this is my address", "Title": "Home Address", "Phone": "05132105521", "Longitude": 0, "Latitude": 0, "Comment": "", "CreateDate": "2016-06-02T14:16:23.507", "UpdateDate": "2016-06-02T14:16:23.507", "Deleted": false, "RowVersion": 10006 }] }</pre>	نمونه محتوای پاسخ دریافتی

سرویس های SaveAddress و DeleteAddress به ترتیب برای ایجاد آدرس جدی برای شخص و یا حذف آدرس آن شخص کاربرد دارد

http://bazaraservices.mahaksoft.com/sync/SaveAddress	url سرویس
Post	نوع متد
<pre>{ "systemSyncID": 1, "userToken": "R9qQvB5_f9izMyVj68XgQosNTe-3ug", "AddressID": 0, "PersonID": 2, "CityCode": 102000, "PostalCode": "1234567890", "AddressLabel": "-----", "Title": "Good Address", "Phone": "05132105521", "Longitude": 0, "Latitude": 0, "Comment": "", "AddressCode": null }</pre>	نمونه محتوای درخواست post
فرمت استاندارد ثبت اشیاء	نمونه محتوای پاسخ دریافتی

سرویس GetGroups برای دریافت لیست گروه های کاربران کاربرد دارد

http://bazaraservices.mahaksoft.com/sync/GetAddresses	url سرویس
Get, Post	نوع متد
<pre>{ "result": "success", "code": 0, "message": "", "data": [{ "PersonGroupID": 1, "Code": 1, "Title": "مشتریان عمومی", "PriceLevel": 1, "CreateDate": "2016-06-02T14:16:23.507", "UpdateDate": "2016-06-02T14:16:23.507", "RowVersion": 10010, "Deleted": false }] }</pre>	نمونه محتوای پاسخ دریافتی

سرویس های ثبت و دریافت اطلاعات سفارش و خرید

از این سرویس ها برای ثبت اطلاعات سفارشات ثبت شده و پرداخت های آنلاین انجام شده در سایت استفاده می شود

url سرویس	توضیحات
/Sync/GetOrders	دریافت لیست سفارشات ثبت شده در سرور. به طور معمول این سرویس در سمت افزونه بازارا فراخوانی می شود، ولی اگر سایت نیاز داشته باشد که وضعیت سفارشات بعد از ثبت آنها را بررسی کند می تواند از این سرویس استفاده کند
/Sync/SaveOrder	ثبت اطلاعات عمومی سفارش
/Sync/DeleteOrder	حذف سفارش
/Sync/GetOrderItems	دریافت لیست اقلام موجود در یک سفارش
/Sync/SaveOrderItem	ثبت اقلام موجود در یک سفارش
/Sync/DeleteOrderItem	حذف اقلام موجود در سفارش
/Sync/GetOrderExtraCosts	دریافت لیست هزینه های جانبی سفارش. مانند هزینه ارسال و یا بسته بندی
/Sync/SaveOrderExtraCost	ثبت هزینه جانبی برای سفارش
/Sync/DeleteOrderExtraCost	حذف هزینه جانبی سفارش
/Sync/GetTransactions	دریافت لیست تراکنش ها (پرداخت های آنلاین صورت گرفته).
/Sync/SaveTransaction	ثبت تراکنش
/Sync/DeleteTransaction	حذف تراکنش
/Sync/CreateNewOrder	سرویس تجمعی ثبت سفارش شامل کلیه اطلاعات سفارش

نکته مهم: به طور معمول برای ثبت سفارش و خرید جدید در سایت فقط لازم است از سرویس `CreateNewOrder` استفاده کنید. با فراخوانی این سرویس کلیه اطلاعات یک سفارش ثبت خواهد شد و ضمناً این سرویس کلیه کنترل های لازم و رعایت کلیدهای خارجی و اصلی در دیتابیس و ... را انجام می دهد و نیاز به ثبت دستی آنها نیست. این سرویس اطلاعات شخص جدید سفارش دهنده را نیز ثبت می کند و بنابراین نیاز به استفاده از سرویس `CreatePerson` نیز نخواهید داشت. علاوه بر همه اینها فراخوانی این سرویس از فراخوانی چندین باره سایر سرویس ها برای ثبت سفارش سریع تر است. بنابر این در قسمت زیر فقط این سرویس به طور کامل توضیح داده خواهد شد.

سرویس /Sync/CreateNewOrder برای ثبت خریدها وسفارشات جدید انجام شده در وب سایت :

http://bazaraservices.mahaksoft.com/sync/CreateNewOrder	url سرویس
Post	نوع متد
<pre>{ "systemSyncID": 1, "userToken": "R9qQvB5_f9izMyVj68XgQosNTe-3ugiG4nOsMS", "Person": { "PersonID": 0, "FirstName": "Javad", "LastName": "Javadi", ..., "Address": { "AddressLabel": "this is my address", "CityCode": 102000, ... } }, "ShippingAddress": { "AddressID": 0, "CityCode": 102000, "PostalCode": "1234567890", "AddressLabel": "-----", "Title": "Good Address", "Phone": "05132105521", "Longitude": 0, "Latitude": 0, "Comment": "" }, "Order": { "OrderDate": "2016-02-17T16:50:24.257", "Status": 0, "CustomerComment": "some customer comment", "Comment": "some comment", "PaymentType": 0 }, "Items": [{ "Quantity": 2, "UnitPrice": 1000, "UnitDiscount": 110, "TotalDiscount": 220, "ProductPriceID": 1, "PriceLevel": 1, "Comment": "some comment" }] }</pre>	نمونه محتوای درخواست post

<pre> "ExtarCosts": [{ "Amount": 123000, "Type": 1, "Description": "some description" }], "Transactions": [{ "DebtAmount": 0, "CreditAmount": 5500, "TransactionType": 0, "ReferenceNumber": "2434252523523525", "TransactionDate": "2016-02-17T16:50:24.257", "Account": "mellat:242425252525", "Comment": "comment" }] } </pre>	
<pre> { "result": "success", "data": { "PersonID": 24, "AddressID": 27, "ShippingAddressID": 27, "OrderID": 3, "OrderItemIDs": [3], "ExtraCostIDs": [2, 3], "TransactionIDs": [] } } </pre>	<p>نمونه محتوای پاسخ دریافتی</p>

آبجکت Person برای ثبت اطلاعات شخص سفارش دهنده استفاده می شود. فرمت اطلاعات این شی دقیقاً مثل ورودی سرویس SavePerson است.

نکته : در صورتی که سفارش یک شخص موجود را ثبت می کنید فقط باید فیلد PersonID آن را مقدار بدهید. در این حالت از سایر فیلدها استفاده ای نمی شود. اگر شخص یک کاربر جدید است PersonID را برابر با - قرار دهید و سایر فیلدهای لازم را پر نمایید.

شی ShippingAddress برای ثبت آدرس مقصد مشتری استفاده می شود. توجه کنید که یک شخص می تواند چندین آدرس داشته باشد و در هر سفارش می تواند آدرس متفاوتی برای ارسال بسته اش ایجاد کند. همانند مورد قبلی اگر

مشتری یکی از آدرس های موجود قبلی اش را برای ارسال بسته انتخاب کرده باشد فقط باید فیلد AddressID را پر نمایید و سایر فیلدها کاربردی نخواهند داشت.

اگر مقدار فیلد ShippingAddress را برابر با null قرار دهید در این صورت از آدرس پیش فرض فرد استفاده خواهد شد. اگر در این حالت شخص یک کاربر جدید بوده آدرس پیش فرض آن همان آدرسی خواهد بود که در شی Person ساخته است.

شی Order برای ثبت اطلاعات عمومی سفارش استفاده می شود.

فیلد Status وضعیت سفارش را مشخص می کند که در هنگام ثبت سفارش جدید مقدار ۰ می گیرد. در سمت افزونه بازارا در نرم افزار حسابداری اگر سفارش تایید و تبدیل به فاکتور شود این فیلد مقدار یک خواهد گرفت و در صورت حذف و یا لغو مقدار ۲ خواهد گرفت.

فیلد PaymentType نوع تسویه حساب سفارش را مشخص می کند. در حال حاضر فقط مقدار ۱ برای پرداخت آنلاین معتبر است.

آرایه Items شامل اقلام موجود در سفارش است که برای هر قلم اطلاعات زیر ثبت می شود.

"Quantity": → تعداد درخواستی
"UnitPrice": → قیمت واحد آن کالا
"UnitDiscount": → تخفیف واحد آن کالا
"TotalDiscount": → تخفیف کل ردیف
"ProductPriceID": → شناسه جزئیات کالای درخواستی
"PriceLevel": → سطح قیمت استفاده شده در این سفارش
"Comment": → توضیحات مشتری

آرایه ExtraCosts شامل هزینه های جانی این سفارش است که نوع هزینه جانبی با فیلد Type مشخص می شود. در حال حاضر فقط مقدار ۱ برای هزینه ارسال در نظر گرفته شده است.

آرایه Transactions برای ثبت تراکنش های مالی صورت گرفته برای این سفارش استفاده می شود. در حال حاضر فقط خرید از طریق پرداخت آنلاین استفاده شده است ولی در آینده از آن برای سیستم های اعتباری نیز می تواند استفاده شود

"DebtAmount": → فعلا استفاده نمی شود
"CreditAmount": → مبلغ پرداخت شده به صورت آنلاین
"TransactionType": → فعلا فقط مقدار ۰ برای پرداخت آنلاین استفاده می شود
"ReferenceNumber": → شماره مرجع بانک برای پرداخت آنلاین
"TransactionDate": "2016-02-17T16:50:24.257",
"Account": "mellat:242425252525" → این شماره حساب را باید قبلا در نرم افزار حسابداری در لیست بانک ها تعریف کرده باشید