



مستندات API سرویس

D A D E H F A . C O M



این مستند توسط شرکت ترنج رهیافت نسل جدید (داده‌فا) Dadehfa.com تهیه و با هدف راهنمایی شرکت‌ها و توسعه‌دهندگان همکار در جهت استفاده از سرویس API سایت‌های فروشگاهی داده‌فا ارائه می‌گردد.

با استفاده از این API، شما قادر خواهید بود تا به سادگی اطلاعات مختلفی از فروشگاه دریافت یا ارسال نمایید. برخی از قابلیت‌های کلیدی این سرویس عبارتند از:

- دریافت لیست محصولات
- دریافت جزئیات هر محصول
- دریافت تصاویر مرتبط با محصولات
- بررسی موجودی کالاها
- دریافت دسته‌بندی‌های محصولات
- ثبت اطلاعات مانند فاکتور فروش

آدرس پایه سرویس (Base URL)

<https://hubapi.dadefaapi.ir/>

در ادامه این مستند، روش استفاده از هر یک از متدهای API به همراه توضیحات کامل، فرمت ورودی و خروجی، نمونه درخواست (cURL)، و نکات مربوط به احراز هویت یا محدودیت‌ها آورده شده است.

آدرس:

GET https://hubapi.dadefapi.ir/api/Product/GetProducts

توضیحات

این متد جهت دریافت لیست آخرین محصولات استفاده می‌شود. اطلاعات به صورت ترتیبی و بر اساس جدیدترین تغییرات (بر مبنای tsCodeId) برگردانده می‌شوند.

پارامترها

Header الزامی:

نام هدر	مقدار	توضیح
Appld	api_key {Base64(ApiKey)}	کلید احراز هویت. مقدار باید با api_key شروع شده و سپس یک رشته base64 از ApiKey (مثلاً شماره سریال) قرار گیرد. نمونه: ==api_key OUM0M0E1NzI4NjVBNTIxMg

Query String الزامی:

نام پارامتر	نوع	مقدار پیش فرض	توضیح
tsCodeId	عدد صحیح (long)	0	شناسه نسخه زمانی. در بار اول باید مقدار 0 ارسال شود تا ۱۰۰ رکورد اول دریافت گردد. در پاسخ، فیلد TsCodeId برای هر محصول ارسال می‌شود که باید بیشینه آن در درخواست بعدی استفاده شود.

نمونه cUrl:

```
curl -X GET \
--header 'Accept: application/json' \
--header 'AppId: api_key OUM0M0E1NzI4NjVBNTIxMg==' \
'https://hubapi.dadefapi.ir/api/Product/GetProducts?tsCodeId=0'
```

نمونه پاسخ (Response)

```
{
  "ResultCode": 200,
  "ResultMessage": "اطلاعات با موفقیت ارسال شد",
  "Result": [
    {
      "ProductId": 1282,
      "ProductName": "کفش زنانه نایک توربو",
      "TsCodeId": 44743141,
      "IsActive": true,
      "IsSalable": true,
      "IsDeleted": false,
      "Vat": 0,
      "GroupId": 68,
      "Brand": null
    },
    {
      "ProductId": 1283,
      "ProductName": "کفش زنانه نایک توربو",
      "TsCodeId": 44743142,
      "IsActive": true,
      "IsSalable": true,
      "IsDeleted": false,
      "Vat": 0,
      "GroupId": 68,
      "Brand": null
    },
    {
      "ProductId": 1284,
      "ProductName": "کفش زنانه نایک توربو",
      "TsCodeId": 44743143,
      "IsActive": true,
      "IsSalable": true,
      "IsDeleted": false,
      "Vat": 0,
      "GroupId": 68,
      "Brand": null
    },
    {
      "ProductId": 1285,
      "ProductName": "کفش زنانه نایک توربو",
      "TsCodeId": 44743144,
      "IsActive": true,
      "IsSalable": true,
      "IsDeleted": false,
      "Vat": 0,
      "GroupId": 68,
      "Brand": null
    }
  ]
}
```

آدرس:

GET https://hubapi.dadefaapi.ir/api/Product/GetVariants

توضیحات

از این متد برای دریافت اطلاعات مربوط به وریانت‌های محصولات استفاده می‌شود. وریانت‌ها شامل مشخصاتی مانند رنگ، سایز، قیمت، موجودی و ... هستند. این اطلاعات بر اساس نسخه زمانی (tsCodeId) فیلتر شده و تنها اطلاعات جدیدتر نسبت به مقدار ارسال‌شده، برگردانده می‌شود.

پارامترها

Header الزامی:

نام هدر	مقدار	توضیح
AppId	api_key {Base64(ApiKey)}	کلید احراز هویت. مقدار باید با api_key شروع شده و سپس یک رشته base64 از ApiKey (مثلاً شماره سریال) قرار گیرد. نمونه: ==api_key OUM0M0E1NzI4NjVBNTIxMg

Query String الزامی:

نام پارامتر	نوع	مقدار پیش فرض	توضیح
tsCodeId	عدد صحیح (long)	0	شناسه نسخه زمانی. در بار اول باید مقدار 0 ارسال شود تا ۱۰۰ رکورد اول دریافت گردد. در پاسخ، فیلد tsCodeId برای هر وریانت ارسال می‌شود که باید بیشینه آن در درخواست بعدی استفاده شود.

نمونه درخواست (cURL)

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'AppId: api_key OUM0M0E1NzI4NjVBNTIxMg==' \
  'https://hubapi.dadefaapi.ir/api/Product/GetVariants?tsCodeId=0'
```

```
{
  "ResultCode": 200,
  "ResultMessage": "اطلاعات با موفقیت ارسال شد",
  "Result": [
    {
      "VariantId": 1314,
      "ProductId": 1282,
      "TsCodeId": 41535119,
      "IsDeleted": false,
      "Price": 298000000,
      "Reduction": 0,
      "ItemBarcode": "1101-11511",
      "Qty": 0,
      "Attributes": [
        {
          "Key": "رنگ",
          "Value": "سبز"
        },
        {
          "Key": "سایز",
          "Value": "37.5"
        }
      ]
    }
  ]
}
```

آدرس:

GET <https://hubapi.dadefaapi.ir/api/Product/GetImages>

توضیحات

این متد جهت دریافت تصاویر مرتبط با محصولات یا وریانت‌ها استفاده می‌شود. نوع داده‌ی برگشتی بسته به پارامتر **type** مشخص می‌شود.

همچنین مانند سایر متدهای این کنترلر، پارامتر **tsCodeId** برای دریافت داده‌های جدیدتر نسبت به نسخه قبلی استفاده می‌شود.

پارامترها

Header الزامی:

نام هدر	مقدار	توضیح
AppId	api_key {Base64(ApiKey)}	کلید احراز هویت. مقدار باید با api_key شروع شده و سپس یک رشته base64 از ApiKey (مثلاً شماره سریال) قرار گیرد. نمونه: ==api_key OUM0M0E1NzI4NjVBNTIxMg

Query String الزامی:

نام پارامتر	نوع	مقدار پیش فرض	توضیح
tsCodeId	عدد صحیح (long)	0	شناسه نسخه زمانی. در بار اول باید مقدار 0 ارسال شود تا 100 رکورد اول دریافت گردد. در پاسخ، فیلد TsCodeId برای هر تصویر ارسال می‌شود که باید ببیشینه آن در درخواست بعدی استفاده شود. لازم به ذکر است که باید tsCodeId را به صورت جداگانه برای تصاویر نوع product و variant ذخیره و استفاده کنید.
type	رشته (string)	product	تعیین نوع تصویر. مقدار می‌تواند product یا variant باشد. در صورت ارسال مقدار اشتباه، پاسخی بازگردانده نمی‌شود.

نمونه درخواست (cURL)

دریافت تصاویر وریانت‌ها:

```
curl -X GET \  
  --header 'Accept: application/json' \  
  --header 'AppId: api_key OUM0M0E1NzI4NjVBNTIxMg==' \  
  
'https://hubapi.dadefaapi.ir/api/Product/GetImages?tsCodeId=0&type=variant'
```

دریافت تصاویر محصولات:

```
curl -X GET \  
  --header 'Accept: application/json' \  
  --header 'AppId: api_key OUM0M0E1NzI4NjVBNTIxMg==' \  
  'https://  
hubapi.dadefaapi.ir/api/Product/GetImages?tsCodeId=0&type=product'
```



```
{
  "ResultCode": 200,
  "ResultMessage": "اطلاعات با موفقیت ارسال شد",
  "Result": [
    {
      "ImageId": 1034,
      "TsCodeId": 14002,
      "IsDeleted": false,
      "Url": "https://cdn.dadehfa.ir/144/barcodes/84cc7bb4-6c9c-41b1-8a5a-40a2710bd877.jfif",
      "VariantId": 1427
    },
    {
      "ImageId": 1035,
      "TsCodeId": 14003,
      "IsDeleted": false,
      "Url": "https://cdn.dadehfa.ir/144/barcodes/c8b2d7ea-549c-415f-83b2-529bad6e7c3a.jpg",
      "VariantId": 1427
    },
    {
      "ImageId": 10311,
      "TsCodeId": 14009,
      "IsDeleted": false,
      "Url": "https://cdn.dadehfa.ir/144/barcodes/9753ea66-5a3d-4794-947a-59cb61984dbe.jpg",
      "VariantId": 1353
    },
    {
      "ImageId": 10312,
      "TsCodeId": 14010,
      "IsDeleted": false,
      "Url": "https://cdn.dadehfa.ir/144/barcodes/6aea0d2e-bf4a-4035-844e-c30dc6adad79.jpg",
      "VariantId": 1353
    }
  ]
}
```

GET <https://hubapi.dadefaapi.ir/api/Product/GetStocks>

توضیحات

این متد برای دریافت موجودی هر وریانت به تفکیک انبار استفاده می‌شود. داده‌ها شامل شناسه وریانت، شناسه انبار، تعداد موجودی و نسخه زمانی (TsCodeId) هستند. در هر بار فراخوانی، با ارسال مقدار مناسب tsCodeId، تنها تغییرات جدید نسبت به فراخوان قبلی دریافت می‌شود.

پارامترها

Header الزامی:

نام هدر	مقدار	توضیح
AppId	api_key {Base64(ApiKey)}	کلید احراز هویت. نمونه: ==api_key OUM0M0E1NzI4NjVBNTIxMg

Query String الزامی:

نام پارامتر	نوع	مقدار پیش‌فرض	توضیح
tsCodeId	عدد صحیح (long)	0	شناسه نسخه زمانی. در بار اول مقدار 0 ارسال شود تا اولین دسته از اطلاعات بازگردد. در پاسخ، فیلد TsCodeId برای هر آیتم موجود است که باید بیشینه آن در درخواست بعدی استفاده شود.

نمونه درخواست (cURL)

```
curl -X GET \
  --header 'Accept: application/json' \
  --header 'AppId: api_key OUM0M0E1NzI4NjVBNTIxMg==' \
  'https://hubapi.dadefaapi.ir/api/Product/GetStocks?tsCodeId=0'
```

نمونه پاسخ (Response)

```
{
  "ResultCode": 200,
  "ResultMessage": "اطلاعات با موفقیت ارسال شد",
  "Result": [
    {
      "VariantId": 1370,
      "TsCodeId": 41699722,
      "WarehouseId": 3,
      "Qty": 1
    },
    {
      "VariantId": 1371,
      "TsCodeId": 41699723,
      "WarehouseId": 3,
      "Qty": 1
    },
    {
      "VariantId": 1374,
      "TsCodeId": 41699726,
      "WarehouseId": 3,
      "Qty": 1
    },
    {
      "VariantId": 3088,
      "TsCodeId": 41699738,
      "WarehouseId": 3,
      "Qty": 1
    }
  ]
}
```

📌 توضیح کلی درباره متد چندمنظوره ValuesController.Post

در این API، متدی به نام Post در کنترلر ValuesController تعریف شده است که از نوع POST بوده و ورودی آن به صورت یک رشته XML است. (string xml) این طراحی به گونه‌ای است که با ارسال XML های مختلف (که شامل کلیدهایی مشخص هستند)، عملکرد متد تغییر می‌کند. یعنی شما با یک مسیر ثابت می‌توانید درخواست‌های متفاوتی ارسال کنید و داده‌های متفاوتی دریافت نمایید.

این ساختار باعث شده که به جای تعریف چندین API جداگانه، از یک متد مرکزی برای چند عملکرد استفاده شود. تصمیم‌گیری در مورد اینکه چه عملیاتی انجام شود، بر اساس محتوای تگ <ReportKey> در XML ارسالی انجام می‌گیرد.

🔧 نکات مهم:

- مسیر متد همیشه ثابت است:

<https://hubapi.dadefaapi.ir/api/Values/Post>

- نوع درخواست POST است.
- مقدار ورودی یک رشته XML بوده که حتماً باید به صورت **Case-Sensitive** نوشته شود (تگ‌ها با دقت به بزرگی و کوچکی حروف).
- در هدر درخواست، مقدار **AppId** برای احراز هویت لازم است.

در ادامه، ما به دو مورد خاص از کاربرد این متد خواهیم پرداخت:

1. دریافت لیست دسته‌بندی‌ها (GetProductGroups) ✓

2. ثبت فاکتور (CreateSaleInvoice) 📄

📌 دریافت لیست دسته‌بندی‌ها (گروه‌های محصولات)

✓ توضیحات کلی

برای دریافت لیست گروه‌های محصولات (دسته‌بندی‌ها) از متد چندمنظوره‌ی POST موجود در کنترلر ValuesController استفاده می‌شود. با ارسال XML مناسب حاوی ReportKey مشخص‌شده، این متد داده‌های مرتبط با گروه‌بندی محصولات را برمی‌گرداند.

🔗 مسیر API

<https://hubapi.dadefaapi.ir/api/Values/Post>

📄 نوع درخواست

POST

✉️ هدرهای لازم

نام هدر	مقدار	توضیح
AppId	api_key {Base64(ApiKey)}	کلید احراز هویت. نمونه: ==api_key OUM0M0E1NzI4NjVBNTIxMg

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'AppId: api_key OUM0M0E1NzI4NjVBNTIxMg==' \
  -d
  '"<root><ReportKey>GetProductGroups</ReportKey><TSCodeID>0</TSCodeID></root>"' \
  'https://hubapi.dadefaapi.ir/api/Values/Post'
```

توضیح ورودی 📄

نام فیلد	نوع داده	توضیحات
ReportKey	string	کلید تعیین نوع گزارش؛ برای دریافت گروه‌ها مقدار آن باید GetProductGroups باشد.
TSCodeID	long	عدد صحیح برای دریافت فقط داده‌هایی که بعد از این شناسه تغییر کرده‌اند. اگر مقدار 0 ارسال شود، کل داده‌ها دریافت می‌شوند.

⚠ توجه: ساختار XML حساس به بزرگی و کوچکی حروف (Case-Sensitive) است.

نمونه ورودی (XML) 📄

```
curl -X POST \
  --header 'Content-Type: application/json' \
  --header 'Accept: application/json' \
  --header 'AppId: api_key OUM0M0E1NzI4NjVBNTIxMg==' \
  -d
  '"<root><ReportKey>GetProductGroups</ReportKey><TSCodeID>0</TSCodeID></r
oot>"' \
  'https://hubapi.dadefaapi.ir/api/Values/Post'
```

نمونه خروجی (JSON) 📄

```
[
  {
    "GroupID": 56,
    "GroupName": "مت یوگا",
    "ParentGroupID": 38,
    "GroupLevel": 2,
    "GroupDiscount": 0,
    "Tags": "",
    "TsCodeID": 41488030,
    "OnlinePriority": 0,
    "IsOnlineActive": false
  },
  {
    "GroupID": 57,
    "GroupName": "کش لوپ",
    "ParentGroupID": 38,
    "GroupLevel": 2,
    "GroupDiscount": 0,
    "Tags": "",
    "TsCodeID": 41488031,
    "OnlinePriority": 0,
    "IsOnlineActive": false
  },
  {
    "GroupID": 58,
    "GroupName": "کش پیلاتس",
    "ParentGroupID": 38,
    "GroupLevel": 2,
    "GroupDiscount": 0,
    "Tags": "",
    "TsCodeID": 41488032,
    "OnlinePriority": 0,
    "IsOnlineActive": false
  }
]
```

ثبت فاکتور فروش(CreateSaleInvoice)

توضیحات کلی

برای ثبت یک فاکتور فروش، از متد POST چندمنظوره موجود در کنترلر ValuesController استفاده می‌شود. این متد با ارسال XML ورودی مناسب و تعیین ReportKey برابر با CreateSaleInvoice، اطلاعات کامل یک فاکتور را دریافت و در سیستم ثبت می‌کند.

این متد از چندین تگ و مجموعه داده استفاده می‌کند که شامل اطلاعات مشتری، آدرس، تاریخ فاکتور، اقلام سبد خرید، پرداخت‌ها و ... می‌باشد. همچنین لازم است برخی از فیلدها مانند StateID و SaleType همیشه مقدار مشخصی داشته باشند.

مسیر API

<https://hubapi.dadefaapi.ir/api/Values/Post>

نوع درخواست

POST

هدرهای لازم

نام هدر	مقدار	توضیح
AppId	api_key {Base64(ApiKey)}	کلید احراز هویت. نمونه: ==api_key OUM0M0E1NzI4NjVBNTIzMg

نمونه درخواست (cURL)

```

curl -X POST \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'AppId: api_key OUMOM0E1NzI4NjVBNTIxMg==' \
-d '"<root><ReportKey>CreateSaleInvoice</ReportKey> \
<SaleInvoice> \
  <SaleDate>2020-12-16</SaleDate> \
  <Mobile>09158128994</Mobile> \
<CouponSerial>سریال کوپن</CouponSerial> \
  <StateID>0</StateID> \
  <WareHouseID>10</WareHouseID> \
  <CustomerAddress>Mashhad</CustomerAddress> \
  <SaleDescription>customer description here</SaleDescription> \
  <DeliveryDate>2020-12-16</DeliveryDate> \
  <SaleType>1</SaleType> \
  <InvoiceItem> \
    <SaleInvoiceID>1</SaleInvoiceID> \
    <SalePrice>150000</SalePrice> \
    <Qty>5</Qty> \
    <ItemName>کفش ورزشی</ItemName> \
    <Vat>9.0</Vat> \
    <Reduction>5</Reduction> \
    <ItemBarcode>626069865656</ItemBarcode> \
  </InvoiceItem> \
  <InvoiceItem> \
    <SaleInvoiceID>1</SaleInvoiceID> \
    <SalePrice>150000</SalePrice> \
    <Qty>5</Qty> \
    <ItemName>کفش ورزشی</ItemName> \
    <Vat>9.0</Vat> \
    <Reduction>5</Reduction> \
    <ItemBarcode>626069865656</ItemBarcode> \
  </InvoiceItem> \
  <InvoiceItem> \
    <SaleInvoiceID>1</SaleInvoiceID> \
    <SalePrice>150000</SalePrice> \
    <Qty>5</Qty> \
    <ItemName>کفش ورزشی</ItemName> \
    <Vat>9.0</Vat> \
    <Reduction>5</Reduction> \
    <ItemBarcode>626069865656</ItemBarcode> \
  </InvoiceItem> \
  <InvoicePayment> \
    <PayPrice>50000</PayPrice> \
    <SalePaymentTypeID>1</SalePaymentTypeID> \
    <PayDescription>توضیحات پرداخت</PayDescription> \
  </InvoicePayment> \
  <InvoicePayment> \
    <PayPrice>70000</PayPrice> \
    <SalePaymentTypeID>2</SalePaymentTypeID> \
    <PayDescription>توضیحات پرداختی مختلف</PayDescription> \
  </InvoicePayment> \
</SaleInvoice> \
</root>"' \
'https://hubapi.dadefapi.ir/api/Values/Post'

```


توضیحات	مقدار مورد انتظار	تگ XML
همیشه باید مقدار ۰ باشد.	0	<StateID>
همیشه باید مقدار ۱ باشد.	1	<SaleType>
باید مقدار بارکد معتبر یکی از محصولات باشد.	-	<ItemBarcode>

 نمونه ورودی (XML)

```
"<root><ReportKey>CreateSaleInvoice</ReportKey>
<SaleInvoice>
  <SaleDate>2020-12-16</SaleDate>
  <Mobile>09158128994</Mobile>
  <CouponSerial>سریال کوپن</CouponSerial>
  <StateID>0</StateID>
  <WareHouseID>10</WareHouseID>
  <CustomerAddress>Mashhad</CustomerAddress>
  <SaleDescription>customer description here</SaleDescription>
  <DeliveryDate>2020-12-16</DeliveryDate>
  <SaleType>1</SaleType>
  <InvoiceItem>
    <SaleInvoiceID>1</SaleInvoiceID>
    <SalePrice>150000</SalePrice>
    <Qty>5</Qty>
    <ItemName>کفش ورزشی</ItemName>
    <Vat>9.0</Vat>
    <Reduction>5</Reduction>
    <ItemBarcode>626069865656</ItemBarcode>
  </InvoiceItem>
  <InvoiceItem>
    <SaleInvoiceID>1</SaleInvoiceID>
    <SalePrice>150000</SalePrice>
    <Qty>5</Qty>
    <ItemName>کفش ورزشی</ItemName>
    <Vat>9.0</Vat>
    <Reduction>5</Reduction>
    <ItemBarcode>626069865656</ItemBarcode>
  </InvoiceItem>
  <InvoicePayment>
    <PayPrice>50000</PayPrice>
    <SalePaymentTypeID>1</SalePaymentTypeID>
    <PayDescription>توضیحات پرداخت</PayDescription>
  </InvoicePayment>
  <InvoicePayment>
    <PayPrice>70000</PayPrice>
    <SalePaymentTypeID>2</SalePaymentTypeID>
    <PayDescription>توضیحات پرداختی مختلف</PayDescription>
  </InvoicePayment>
</SaleInvoice>
</root>"
```

 نمونه خروجی در صورت خطا در بارکدها

```
[
  {
    "Result": "error",
    "MessageText": "بارکدهای نامعتبر که در لیست شما وجود دارد",
    "Data":
    "<root><InvoiceItem><SalePrice>150000.00</SalePrice><Qty>5.000</Qty><ItemName> کفش ورزشی</ItemName><Vat>9</Vat><Reduction>5.000</Reduction><ItemBarcode>626069865656</ItemBarcode></InvoiceItem><InvoiceItem><SalePrice>150000.00</SalePrice><Qty>5.000</Qty><ItemName> کفش ورزشی</ItemName><Vat>9</Vat><Reduction>5.000</Reduction><ItemBarcode>626069865656</ItemBarcode></InvoiceItem><InvoiceItem><SalePrice>150000.00</SalePrice><Qty>5.000</Qty><ItemName> کفش ورزشی</ItemName><Vat>9</Vat><Reduction>5.000</Reduction><ItemBarcode>626069865656</ItemBarcode></InvoiceItem></root>"
  }
]
```

✓ نمونه خروجی در صورت موفقیت

```
[
  {
    "Result": "ok",
    "MessageText": 1044
  }
]
```

⚠ نکته مهم

ممکن است وضعیت پاسخ (HTTP Status Code) همیشه 200 OK باشد، حتی در صورتی که درخواست با خطا مواجه شده باشد. بنابراین حتماً باید بدنه پاسخ (Result) بررسی شود:

- اگر $\text{Result} = \text{ok}$ → درخواست با موفقیت انجام شده است.
- اگر $\text{Result} = \text{error}$ → درخواست با خطا مواجه شده است و باید جزئیات بررسی شود.