

راهنمای وب سرویس تدبیر ۸

فهرست:

۳.....	معرفی وب سرویس.....
۴.....	قابلیت‌های نرم‌افزار وب سرویس تدبیر.....
۵.....	نحوه ایجاد لینک برای پرس‌وجوها.....
۷.....	پارامترها.....
۱۵.....	مدیریت دسترسی‌ها.....
۱۷.....	توکن حساب کاربری.....
۲۱.....	پلاگین.....
۲۶.....	نحوه ارسال و دریافت اطلاعات در وب سرویس.....
۲۸.....	نمونه کد فراخوانی وب سرویس.....

نرم افزار وب سرویس تدبیر این امکان را ایجاد می کند که سایر نرم افزارها بتوانند با نرم افزار تدبیر ارتباط برقرار کنند. وب سرویس تدبیر، یک نرم افزار بر پایه سرویس Rest می باشد که با کمک پروتکل HTTP ارتباط بین سرویس دهنده (نرم افزار تدبیر) و سرویس گیرنده (نرم افزار کاربر) را در محیط وب فراهم می سازد. در پروتکل HTTP می توان با کمک دستورات وب مانند Get ، Post ، Put و Delete وب سرویس را فراخوانی کرد. بسیاری از نرم افزارهای تحت وب مانند مرورگرها، نرم افزارهای اندروید، نرم افزار آفیس و... قابلیت استفاده از وب سرویس را دارند.

- رابط کاربری تحت وب
- امکان قرار دادن دستورات SQL در برنامه بجای استفاده از Store Procedure
- امکان ایجاد ارتباط با چندین پایگاه داده و فراخوانی آنها با استفاده از نام‌های تعریف شده توسط کاربر
- امکان ایجاد پارامتر برای درخواست‌ها
- امکان ایجاد پارامتر در سطح سرویس و ارتباط با پارامترهای SQL
- اعتبارسنجی پارامترها
- امکان تبدیل کدپیچ تدبیر
- امکان تست توابع ایجاد شده
- گزارش‌گیری رویدادها
- امکان ورود و خروج اطلاعات برنامه
- امکان انتقال اطلاعات برنامه از نسخه ۲,۰,۰,۶ به نسخه‌های بعدی
- امکان ایجاد پلاگین برای امکاناتی که قابل پیاده‌سازی در سرویس نمی‌باشد
- استفاده از setup برای نسخه‌های مختلف برنامه و نصب آسان برنامه
- مدیریت آدرس‌ها URL
- امکان استفاده از انواع داده‌های قابل انتقال در محیط وب
- مدیریت کاربران
- امکان استفاده از توکن امنیتی
- امکان ارسال پارامترهای برنامه مانند نام کاربری و ... به پرس‌وجوها
- امکان تست پلاگین‌ها
- امکان بازیابی رمزعبور
- امکان تعریف متادیتای سرویس در فایل پلاگین
- گزارش دریافت و ارسال‌ها

نحوه ایجاد لینک برای پرس و جوها

لینک های پرس و جو

برای فراخوانی وب سرویس، به هر پرس و جو یک مسیر نسبت داده می شود. توسط آدرس این مسیر می توان پارامترها را به وب سرویس ارسال، پرس و جوها را اجرا و مقدار آن را دریافت کرد. در حالتی که برای فراخوانی پرس و جو نیاز به استفاده از توکن است، قبل از اجرای پرس و جو مراحل لازم در قسمت نحوه استفاده از توکن ها را در راهنمای برنامه مطالعه کنید.

آدرس پیش فرض

از این آدرس دهی می توان در سرویس های پایگاه داده و پلاگین استفاده کرد. در این آدرس دهی باید در فرمهای **سرور، سرویس و پرس و جو** در پنل وب سرویس، مقدار مسیر حتما وارد شده باشد و نام مسیرها، باید تکراری نباشد. برای فراخوانی سرویس پایگاه داده بعد از مسیر سرور و پورت و قبل از آدرس پرس و جو مقدار **db** وارد شود و همچنین برای فراخوانی سرویس پلاگین بعد از مسیر سرور و پورت و قبل از آدرس پرس و جو مقدار **pl** وارد شود.

فرمت آدرس دهی سرویس پایگاه داده بصورت زیر است:



- **آدرس سرور:** مسیر سروری است که وب سرویس بروی آن نصب شده است.
- **پورت:** شماره درگاه وب سرویس می باشد. این مقدار در هنگام نصب نرم افزار توسط کاربر مقداردهی شده است.
- **مسیر سرور:** مقدار فیلد **نام مسیر** در فرم سرور، اگر این مقدار ارسال نشود از سرور پیش فرض در استفاده می شود.
- **سرویس:** مقدار فیلد **نام مسیر** در فرم سرویس
- **پرس و جو:** مقدار فیلد **نام مسیر** در فرم پرس و جو
- پارامترهای ارسالی به سرور
- **پارامترهای ۱:** فیلد **نام پارامتر** در آدرس در فرم پارامترهای پرس و جو
- **مقدار:** این مقدار توسط کاربر وارد می شود.

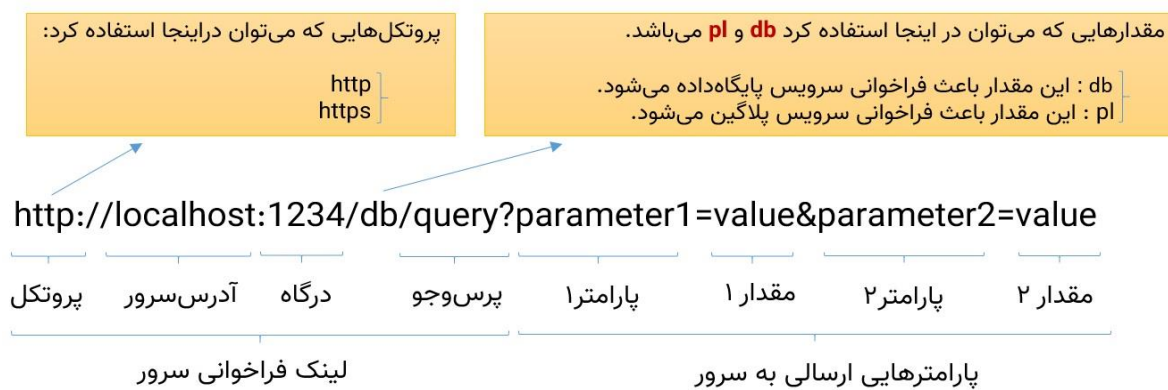
- برای سایر پارامترها، مشابه بالا می‌باشد.

آدرس پیش‌فرض (استفاده از سرور و سرویس پیش‌فرض)

از این آدرس‌دهی می‌توان در سرویس‌های پایگاه‌داده و پلاگین استفاده کرد. در این آدرس‌دهی باید در فرمهای سرور، سرویس و پرس‌وجو در پنل وب سرویس، مقدار نام‌مسیر حتما وارد شده باشد و نام مسیرها باید تکراری نباشد.

برای فراخوانی سرویس پایگاه‌داده بعد از مسیر سرور و پورت و قبل از آدرس پرس‌وجو مقدار db/ وارد شود و همچنین برای فراخوانی سرویس پلاگین بعد از مسیر سرور و پورت و قبل از آدرس پرس‌وجو مقدار pl/ وارد شود. برای حذف سرویس و سرور در لینک، باید سرویس و دیتابیس پیش‌فرض انتخاب شده باشد، در غیر این‌صورت پیغام خطا بازگردانده می‌شود.

فرمت لینک سرور در حالتی که از سرور و سرویس پیش‌فرض استفاده شده است، بصورت زیر است:



- آدرس سرور: مسیر سروری است که وب سرویس بروی آن نصب شده است.

- پورت: شماره درگاه وب سرویس می‌باشد. این مقدار در هنگام نصب نرم‌افزار از کاربر درخواست شده است.

- پرس‌وجو: مقدار فیلد نام مسیر در فرم پرس‌وجو

- پارامترهای ارسالی به سرور

- پارامترهای ۱: فیلد نام پارامتر در آدرس در فرم پارامترهای پرس‌وجو

- مقدار: این مقدار توسط کاربر وارد می‌شود.

- برای سایر پارامترها، مشابه بالا می‌باشد.

معرفی پارامترها

پارامترها این امکان را برای وب سرویس ایجاد می‌کنند که در زمان اجرای پرس‌وجوها می‌توان مقادیر مورد نیاز را به پرس‌وجوها منتقل کرد. پارامترها معمولاً از نوع رشته، عدد صحیح و یا تاریخ هستند. پارامترها می‌توانند دارای مقدار پیش‌فرض، خالی، و یا توسط وب سرویس در زمان اجرا مقداردهی شوند.

پارامترهای سرویس و پارامترهای پرس‌وجو

پارامترها را می‌توان در سرویس و یا در پرس‌وجو تعریف کرد.

- پارامترهای سرویس

- این پارامترها در سطح سرویس قابل تعریف هستند و از آنها می‌توان در پرس‌وجوهای آن سرویس بصورت مشترک استفاده کرد.
- در صورت تغییر مقدار این پارامتر، این تغییر در تمامی پرس‌وجوها اعمال می‌شود.
- پارامترها نمی‌توانند همانام باشند.
- برای استفاده از این پارامترها باید توسط فیلد "پارامتر سرویس مرتبط" در فرم ایجاد / اصلاح پارامترهای کوئری، آنها را به پارامترهای پرس‌وجو مرتبط کرد.

- پارامترهای پرس‌وجو:

- این پارامترها در سطح پرس‌وجو تعریف می‌شوند.
- از این پارامترها فقط در همان پرس‌وجو می‌توان استفاده کرد.
- پارامترهای یک پرس‌وجو نمی‌توانند همانام باشند.

پارامترهای سرویس

در صفحه پایگاه‌داده/سرویس پرس‌وجو با انتخاب یک سرویس و دکمه "پارامترهای سرویس" می‌توان وارد صفحه پارامترهای سرویس شد.

پارامتر سرویس دارای فیلدهای زیر می‌باشد:

- نام: نام پارامتر نمی‌تواند تکراری باشد.
- مقدار: مقداری است که در هنگام اجرای پرس‌وجو از آن استفاده شده و به پرس‌وجو منتقل می‌شود.

- نوع: یک مقدار اختیاری است و می‌تواند همانام تعریف شوند.

پارامترهای پرس‌وجو

در پنل وب سرویس در فرم پرس‌وجو با انتخاب یک پرس‌وجو و دکمه پارامترها می‌توان وارد صفحه پارامترهای پرس‌وجو شد.

فیلدهای پارامتر پرس‌وجو به ترتیب زیر هستند:

- نام: نام پرس‌وجو می‌باشد، باید تکراری نباشد و در آن می‌توان از حروف فارسی استفاده کرد.
- ترتیب پارامترها: برای مرتب‌سازی پارامترها استفاده می‌شود.
- نام پارامتر در آدرس: از این مقدار برای مشخص کردن پارامتر در URL و یا متن درخواست پرس‌وجو استفاده می‌شود.
- نام پارامتر در پرس‌وجو: از این مقدار در داخل متن کوئری و یا نام پارامتر StoreProcedure در sql مورد استفاده قرار می‌گیرد.
- نوع پارامتر: پارامتر می‌تواند توسط سرویس مقداردهی شود، توسط کاربر ارسال شود و یا برای اعتبارسنجی پرس‌وجو مورد استفاده قرار گیرد. در اینجا می‌توان نحوه استفاده از پارامتر را مشخص کرد.
- شرح: توضیحات مربوط به پارامتر توسط کاربر در اینجا شرح داده می‌شود.

فیلد نام پارامتر در آدرس

از این فیلد برای ارسال مقدار به وب سرویس توسط ارسال کننده (نرم‌افزار کاربر) مورد استفاده قرار می‌گیرد. برای ارسال پارامترهای پرس‌وجو به وب سرویس توسط کاربر، پارامترها باید بصورت نام/مقدار ارسال شوند. این فیلد نام ارسالی پارامتر می‌باشد. برای ارسال پارامتر در پروتکل HTTP می‌توان پارامترها را توسط آدرس و همچنین متن درخواست ارسال کرد. نام این فیلد نمی‌تواند در یک پرس‌وجو تکراری باشد.

فیلد نام پارامتر در پرس‌وجو

از این فیلد برای ارسال مقدار به پرس‌وجو توسط وب سرویس مورد استفاده قرار می‌گیرد. از مقدار این پارامتر برای مقداردهی پارامتر همانام در StoreProcedure و یا در query های SQL مورد استفاده قرار می‌گیرد. در کوئری‌های SQL پارامترها بصورت @pParameterName تعریف می‌شوند. در اینجا برای مرتبط کردن پارامتر پرس‌وجو به پارامتر SQL باید این فیلد با pParameterName مقداردهی شود. در این فیلد نباید از کاراکتر @ استفاده شود. مابقی کاراکترها باید مشابه آن در SQL باشد.

برای پارامترهای StoreProcedure ها نیز به روش مشابه پارامترها را در وب سرویس تعریف می‌شود.

انواع پارامترها

- پارامترهایی که توسط وب سرویس مقداردهی می‌شوند.
- پارامترهایی که توسط کاربر در زمان اجرا ارسال می‌شوند.
- پارامترهایی که برای اعتبارسنجی درخواست استفاده می‌شوند.

پارامترهای مقداردهی شده توسط وب سرویس

این نوع از پارامترها توسط وب سرویس مقداردهی می‌شوند و در دسترس کاربر نمی‌باشند.

انواع پارامترها:

- نام کاربری
- پست الکترونیک کاربر
- نام سرور
- نام پایگاه داده
- نام سرویس
- نام پرس‌وجو

پارامترهای قابل ارسال به وب سرویس

این نوع از پارامترها توسط کاربر مقداردهی می‌شوند. در درخواست ارسال شده توسط کاربر، باید مقدار این پارامتر نیز ارسال شود.

در شرایط زیر می‌توان این پارامترها را ارسال نکرد و وب سرویس از مقدار پیش فرض یا بصورت null مقداردهی می‌کند:

- برای این پارامترها مقدار پیش فرض تعریف شده باشد.
- در فرم پارامترها، مقدار آن بتواند خالی باشد، انتخاب شده باشد.
- در فرم پارامترها، مقدار Null بررسی نشود، انتخاب شده باشد.

برای پارامتر ارسال شده توسط کاربر باید نوع سیستمی آن نیز مشخص شود:

- رشته‌ها و کاراکترها -> String
- اعداد صحیح -> Int32
- اعداد اعشاری -> Decimal

- تاریخ -> DateTime

- ...

در صورتیکه مقدار ارسال شده از نوع String باشد، مقدار ارسال شده می تواند Null نباشد ولی رشته کاراکتر خالی باشد. برای رفع این مشکل در فیلد بررسی مقدار Empty برای نوع String می توان مشخص کرد که در صورت خالی بودن رشته کاراکتر، پیغام خطا بازگردانده شود.

در حالتی که نمی خواهیم مقدار Null بودن پارامتر بررسی شود، در فیلد بررسی مقدار Null می توان مشخص کرد که این فیلد می تواند Null باشد و یا بالعکس در صورت Null بودن پیغام خطا بازگردانده شود.

در حالتی که بخواهیم مقدار پارامتر سرویس بجای این پارامتر به وب سرویس ارسال شود، می توان از لیست پارامتر سرویس مرتبط، پارامتر سرویس را انتخاب کنیم. در صورتیکه در درخواست ارسال شده این پارامتر مقداردهی شده باشد، الویت با مقدار ارسال شده توسط کاربر می باشد، و در صورت خالی بودن از این مقدار استفاده خواهد شد.

پارامترهای اعتبارسنجی درخواست

در حالتی که بخواهیم درخواست ارسال شده را اعتبارسنجی کنیم، از این گزینه می توان استفاده کرد. در این گزینه یک مقدار پیش فرض تعریف می شود. مقدار ارسال شده توسط کاربر با مقدار این فیلد مقایسه شده و در صورتیکه دو مقدار مشابه هم باشند، پرس و جو اجرا می شود. در غیر این صورت پیغام خطا بازگردانده خواهد شد. در صورتیکه از این گزینه استفاده شده باشد، ارسال مقدار برای این پارامتر الزامی می باشد. در این حالت مقدار ارسال شده به وب سرویس، به پرس و جو ارسال داده نخواهد شد و فقط برای بررسی در وب سرویس استفاده می شود. می توان بجای مقدار پیش فرض از پارامترهای سرویس استفاده کرد که در این حالت از لیست پارامتر سرویس مرتبط، پارامتر سرویس را باید انتخاب کنید.

نحوه ارسال پارامترها

- داده هایی که توسط کاربر ارسال می شوند بصورت نام/مقدار می باشند.
- برای ارسال داده در HTTP از دو روش می توان استفاده کرد:
- ارسال داده ها توسط آدرس درخواست
 - از این روش می توان در تمامی حالت های Get/Post/Put/Delete استفاده کرد.
- ارسال داده ها توسط متن درخواست
 - از این روش نمی توان در حالت Get استفاده کرد.

ارسال پارمترها توسط آدرس درخواست

در این روش داده ها بصورت نام/مقدار در انتهای آدرس URL برای وب سرویس ارسال می شوند. برای ارسال داده ها، در انتهای آدرس و بعد از قرار دادن کاراکتر ؟ می توان پارامترها را بصورت مقدار= نام ارسال کرد. در حالتی که چند پارامتر را می خواهید ارسال کنید، باید پارامترها را با کاراکتر & از هم جدا کنید. بدلیل محدود بودن طول آدرس، از این حالت فقط برای ارسال محدود از داده ها می توان استفاده کرد.

برای نمونه، در مثال زیر پارامتر نام حساب و کد حساب به همراه مقادیر آن، به وب سرویس ارسال شده است.



ارسال پارامترها توسط متن درخواست

از این حالت نمی توان در Get استفاده کرد. این حالت محدودیت طول داده را ندارد و در صورتیکه طول داده ها زیاد باشد، باید پارامترها را در متن درخواست برای وب سرویس ارسال کرد. برای ارسال داده ها در متن درخواست، باید از یک فرمت استاندارد استفاده شود و همچنین در header درخواست باید نوع فرمت نیز توسط پارامتر `Content-Type` به وب سرویس ارسال شود. از انواع فرمت هایی که می توان استفاده کرد، دو فرمت زیر بیشتر مورد استفاده قرار میگیرد.

- `application/x-www-form-urlencoded`
- `application/json`

برای مثال در درخواست زیر، پارامترها در متن درخواست بافرمت JSON ارسال شده است. همچنین در header درخواست، نوع فرمت توسط پارامتر `Content-Type` برای وب سرویس ارسال شده است.

```
POST http://localhost/db/sample/query2 HTTP/1.1
Host: localhost:80
Content-Type: application/json

{
  "name" : "حساب کاربری 1",
  "code" : "11111",
}
```

در مثال زیر از فرمت application/x-www-form-urlencoded برای ارسال پارامترها استفاده شده است.

```
POST http://localhost/db/sample/query2 HTTP/1.1
Host: localhost:80
Content-Type: application/x-www-form-urlencoded

name=1&code=11111
```

انواع قابل تعریف در پرس‌وجو و پارامترهای پرس‌وجو در وب سرویس

تعریف نوع در پارامترهای ارسالی

این امکان در فرم پارامترهای پرس‌وجو تعریف می‌شود و برای هر پارامتر به صورت جداگانه قابل ایجاد می‌باشد. در وب سرویس فیلدی به نام نوع مقدار پارامتر است که براساس انواع مقدارهای قابل تعریف در فریم ورک ایجاد شده است. این فیلد عملیاتی برای تبدیل نوع پارامتر ورودی انجام نمی‌دهد و صرفاً جهت مقداردهی پرس‌وجو از آن استفاده می‌شود. به دلیل اینکه این فیلد برای مقداردهی در پایگاه داده استفاده می‌شود، اگر نوع آن، متناسب با نوع پارامتر مورد استفاده در دیتابیس نباشد، پیغام خطا بازگردانده می‌شود. برای جلوگیری از خطا می‌توان از جدول زیر برای انتخاب نوع مناسب پارامتر استفاده کرد.

مقدارهای با علامت * در ادامه توضیح داده شده است.

نوع پارامتر در پایگاه داده	نوع پارامتر در وب سرویس
varchar / nvarchar / xml	System.String
bigint / int / tinyint / smallint	System.Int64
char / nchar / text / ntext	System.String
money / numeric / float	System.Double
datetime*	System.String
decimal	System.Decimal
uniqueidentifier*	System.String
varbinary*	System.String

آرایه باینری Byte Array / varbinary / Image

برای پارامترهایی از نوع Byte Array باید قبل از ارسال، تبدیل مقدار به System.String انجام شود، برای مثال می‌توان با کمک تبدیل آرایه بایت به Json آن را بصورت رشته‌ای از کاراکترها به وب سرویس ارسال کرد. با کمک این روش می‌توان فایل‌ها و ... را به عنوان پارامتر ورودی به وب سرویس ارسال کرد. در اینجا فیلد متناسب آن در پایگاه داده varchar / nvarchar و

یا فیلدهای مشابه می‌باشد. برای سایر انواع `ByteArray` مانند بالا، مقدار آرایه باینری به `Base 64 String` تبدیل شود و به صورت `System.String` به وب سرویس ارسال شود و بعد از ارسال به وب سرویس در داخل پرس‌وجو به `varbinary` و یا انواع مشابه آن تبدیل شود.

تاریخ Datetime

نوع فیلد تاریخ وابسته به نوع تعریف شده برای پارامتر ورودی داخل پرس‌وجو در پایگاه‌داده است. برای جلوگیری از خطای تاریخ شمسی به میلادی و یا بالعکس، بهتر است نوع این فیلد `nvarchar` بعنوان پارامتر ورودی پرس‌وجوی پایگاه‌داده تعریف شود و برای تبدیل آن و در داخل پرس‌وجو، نوع آن بررسی و تبدیل لازم انجام شود. در این حالت نوع پارامتر ورودی در فرم پارامترها از نوع `System.String` است.

Guid (uniqueidentifier)

برای ارسال پارامترهای `Guid`، باید آن را بصورت `Sytem.String` به وب سرویس ارسال کرد و تبدیل آن، در داخل پرس‌وجو انجام شود.

انواع قابل تعریف برای پارامترهای دریافتی

در فرم ایجاد پرس‌وجو، توسط فیلدهای **فرمت**، **نوع پاسخ**، **تبدیل مقدار** و **نوع کدپیچ** می‌توان نوع مقدار بازگشتی را مشخص کرد، پارامترهای `header` را تغییر داد و تبدیل‌های لازم را انجام داد.

پارامتر فرمت ایجاد پرس‌وجو

در فیلد فرمت نوع مقدار پاسخ مشخص می‌شود. با کمک این فیلد وب سرویس تبدیل‌های لازم را برای مقدار بازگشتی انجام می‌دهد. مقدارهایی که می‌توان برای این فیلد تعریف کرد:

- **Binary** : مقدار بازگشتی از نوع بایت خواهد بود. اگر در فیلد تبدیل مقدار، نوع آرایه و یا آرایه از اشیاء انتخاب شده باشد، مقدار بازگشتی آرایه‌ای از بایت (`Byte Array`) خواهد بود.
- **Json** : مقدار بازگشتی به `Json` تبدیل می‌شود. این نوع پیش‌فرض وب سرویس است. (مقدار بازگشتی پرس‌وجو باید قابل تبدیل به این نوع باشد و در غیراینصورت خطا بازگردانده می‌شود. مقدار بازگشتی می‌تواند یک `Object` و یا آرایه از اشیا و.. باشد.)
- **Bson** : مقدار بازگشتی به `Bson` تبدیل می‌شود. (مقدار بازگشتی پرس‌وجو باید قابل تبدیل به این نوع باشد و در غیراینصورت خطا بازگردانده می‌شود.)

- **Xml** : مقدار بازگشتی به xml تبدیل خواهد شد. (مقدار بازگشتی پرس و جو باید قابل serialize شدن باشد).
- **Text** : بروی مقدار بازگشتی تبدیلی انجام داده نمی شود. (مقدار پرس و جو نیز باید از این نوع و یا انواع مشابه مانند css,Html و ... باشد).

پارامتر نوع پاسخ ایجاد پرس و جو

پارامتر نوع پاسخ یا همان MIME Type می باشد و در قسمت Content-Type پاسخ وب سرویس به سرویس گیرنده ارسال می شود. سرویس گیرنده با کمک این پارامتر از محتوای ارسال شده مطلع شده و بر اساس آن عملیات مرتبط را انجام می دهد. برای مثال اگر در این قسمت مقدار application/xml وارد شود، در مرورگرها برنامه Xml Editor فعال شده و محتوای بازگشتی در آن نمایش داده می شود.

مقادیری که در اینجا بیشتر مورد استفاده قرار می گیرد:

- Application/json
- Application/xml

برای اینکه پاسخ وب سرویس قابل تبدیل به مقدارهای بالا باشد، در قسمت فرمت باید مقدار متناسب آن نیز انتخاب شود. برای نمونه اگر از مقدار application/json استفاده شده باشد، در پارامتر فرمت نیز باید مقدار Json نیز انتخاب شده باشد.

پارامتر تبدیل مقدار ایجاد پرس و جو

انواع قابل تبدیل در وب سرویس به شرح زیر می باشد:

- **هیچکدام**: هیچ تبدیلی بروی مقدار ورودی انجام نمی شود.
- **آرایه**: مقدار بازگشتی که دارای چندین مقدار مشابه هم باشد مانند لیستی از اعداد، متن و ...
- **اشیاء**: برای مثال می توان هر سطر از داده ها را یک شی فرض کرد. اشیاء بصورت [Name] = [Value] بازگردانده خواهند شد. نام در اینجا، نام ستون جدول است.
- **آرایه ای از اشیاء**: در اینجا می توان هر سطر از داده ها را یک شی و مجموعه ای از سطرها را آرایه ای از اشیاء فرض کرد.

پارامتر نوع کدپیچ ایجاد پرس و جو

برای پایگاه داده هایی که نیاز به تبدیل کدپیچ سامان است با انتخاب این فیلد، تبدیل لازم انجام خواهد شد. همچنین این تبدیل بروی پارامترهای ارسالی نیز انجام می شود. این فیلد دارای تبدیلهای دیگری مانند تبدیل Base 64 String و یا یونیکد نیز است.

کاربران وب سرویس

برای ورود به سیستم و استفاده از امکانات وب سرویس و همچنین اجرای پرس‌وجوها نیاز به ایجاد حساب کاربری می‌باشد. بعد از نصب برنامه، در راه‌اندازی وب سرویس، یک حساب کاربری توسط کاربر ایجاد می‌شود که دسترسی مدیریت سرویس را دارد. می‌توانید با این حساب کاربری وارد سیستم شوید. در صفحه **مدیریت کاربران / کاربران** و با انتخاب دکمه جدید می‌توانید وارد صفحه ایجاد حساب کاربری شوید. برای ایجاد حساب کاربری باید یک آدرس ایمیل که قبلاً با آن حساب کاربری ایجاد نشده باشد، و همچنین نام کاربری غیر تکراری، کاربر جدید را تعریف کنید. **برای استفاده از امکان بازیابی رمز عبور، سرویس پست الکترونیک را باید فعال کنید.**

نقش‌های کاربر

برای استفاده از قسمت‌های مختلف برنامه باید دسترسی‌های لازم به کاربر داده شود. برای دادن این دسترسی‌ها در صفحه **مدیریت کاربران / نقش‌های کاربر** با انتخاب نقش مورد نظر و انتخاب دکمه **کاربران مرتبط** وارد صفحه کاربران مرتبط با آن نقش می‌شوید. در این صفحه می‌توانید کاربر جدید را اضافه کنید.

نقش‌هایی که بیشتر مورد نیاز است:

- اجرای پرس‌وجو: نقش پیش‌فرض برای استفاده از توکن می‌باشد. در صورتیکه کاربر جدید برای اجرای پرس‌وجو نیاز به دسترسی دارد، باید به این نقش اضافه شود. **(کاربری که در هنگام راه‌اندازی وب سرویس ایجاد می‌شود، دارای تمامی دسترسی‌های لازم برای مدیریت سرویس است و بهتر است از آن برای فراخوانی وب سرویس استفاده نکرد. برای فراخوانی سرویس یک کاربر جدید (کاربر سرویس) تعریف شود و فقط دسترسی اجرای پرس‌وجو را به آن اختصاص داده شود.)**

دسترسی‌های پرس‌وجو

هر پرس‌وجو را می‌توان به یک نقش مرتبط کرد. بصورت پیش‌فرض همه پرس‌وجوهای جدید به نقش **اجرای پرس‌وجو** مرتبط هستند. می‌توان دسترسی‌های پرس‌وجو را به دو حالت زیر تغییر داد:

- حالت اول: نقش‌های مرتبط با پرس‌وجو در حالت هیچکدام باشد. در این حالت برای اجرای پرس‌وجو نیازی به دسترسی نیست و همه کاربران سیستم و همچنین درخواست‌های نامشخص می‌توانند پرس‌وجو را اجرا کنند. برای اینکه درخواست ارسال شده را در این حالت اعتبارسنجی کرد، برای پرس‌وجو می‌توان پارامتر اعتبارسنجی تعریف کرد. ایجاد پارامتر اعتبارسنجی در قسمت پارامترها توضیح داده شده است. در این حالت حتی با داشتن این پارامتر، دسترسی کاربر بررسی نمی‌شود.

- حالت دوم: پرس‌وجو به یکی از نقش‌های کاربران مرتبط شده باشد. در این حالت با کمک توکن، کاربری که دارای این نقش است می‌تواند پرس‌وجوی خود را اجرا کند. برای نحوه استفاده از توکن، در قسمت توکن‌های حساب کاربری توضیح داده شده است.

پروتکل HTTP

کد زیر یک ارسال درخواست و پاسخ وب سرویس در پروتکل HTTP می باشد.

ارسال درخواست

```
GET http://localhost/db/sample/query1 HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Host: localhost:80
```



پاسخ وب سرویس

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 16 Sep 2020 12:00:00 GMT
Content-Length: 24

"2020-09-16T12:00:00.00"
```

ارسال درخواست HTTP

در درخواست زیر با کمک دستور Get یک فراخوانی به وب سرویس ارسال شده است. در دستور Get می توان پارامترها را توسط URL ارسال کرد. در سایر دستورات همچنین می توان پارامترها را با کمک متن درخواست، ارسال کرد. در Header درخواست می توان داده های مربوط به نوع درخواست و نحوه دریافت پاسخ را مشخص کرد. برای نمونه در درخواست زیر، در سطر دوم با کمک پارامترهای header نوع قابل قبول از فرمت داده ها (که در اینجا JSON می باشد) از وب سرویس خواسته شده است.

ارسال درخواست به وب سرویس

```
GET http://localhost/db/sample/query1 HTTP/1.1
Accept: application/json
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
Host: localhost:80
```

دریافت پاسخ HTTP

وب سرویس بعد از دریافت درخواست و اجرای پرس و جو، یک پاسخ مشابه زیر را به سرویس گیرنده ارسال می کند. در پاسخ ارسال شده می توان به پارامترهای زیر در header می توان اشاره کرد:

استاندارد وب مورد استفاده که در اینجا HTTP/1.1 می باشد.

نتیجه درخواست که در اینجا ۲۰۰ به معنی موفقیت آمیز بودن اجرا می باشد به سرویس گیرنده ارسال شده است. در صورت خطا کد مربوطه در پاسخ بازگردانده می شود. سایر پارامترها مربوط به مشخصات سرویس، طول متن پاسخ و ... می باشد.

در قسمت پایین بعد از پارامترهای header، نتیجه اجرای پرس و جو توسط وب سرویس بازگردانده می شود. برای نمونه، در پرس و جوی وب سرویس خواسته شده ساعت سیستم در پاسخ بازگردانده شود.

پاسخ وب سرویس

```
HTTP/1.1 200 OK
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/json; charset=utf-8
Expires: -1
Server: Microsoft-IIS/8.5
X-AspNet-Version: 4.0.30319
X-Powered-By: ASP.NET
Date: Wed, 16 Sep 2020 12:00:00 GMT
Content-Length: 24

"2020-09-16T12:00:00.00"
```

حساب کاربری

برای استفاده از حساب کاربری در وب سرویس، نیاز به استفاده از توکن امنیتی می باشد. قبل از فراخوانی سرویس باید یک درخواست Post که حاوی نام کاربری و رمز عبور است، به وب سرویس ارسال شود. بعد از بررسی حساب کاربری توسط وب سرویس، یک توکن امنیتی ایجاد شده و در پاسخ بازگردانده می شود. بعد از دریافت توکن و تا قبل از تاریخ انقضاء توکن، می توان در فراخوانی پرس و جوها از این توکن بجای نام کاربری و رمز عبور استفاده کرد. حساب کاربری باید قبلاً در وب سرویس ایجاد شده باشد و دسترسی های لازم نیز به حساب کاربری اختصاص داده شده باشد. برای جلوگیری از دسترسی به نام کاربری و توکن از SSL استفاده شود.

نحوه دریافت توکن

برای دریافت توکن امنیتی، یک درخواست Post مشابه زیر به وب سرویس ارسال می شود:
http://localhost/db/token بجای http://localhost/ باید مسیر وب سرویس بروی سرور خود را جایگزین کنید. مقدارهایی که باید در متن Post قرار گیرد، بصورت زیر می باشد:

Username -

- Password
- Grant_type

```
POST /db/token HTTP/1.1
Host: localhost
Connection: keep-alive
Accept: */*
X-Requested-With: XMLHttpRequest
Content-Type: application/x-www-form-urlencoded

username=[UserName]&password=[Password]&grant_type=password
```

بعد از ارسال درخواست، وب سرویس بعد از بررسی حساب کاربری و در صورتیکه اطلاعات ارسال شده صحیح باشد، پاسخ مشابه زیر را ارسال می کند:

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Date: Wed, 16 Sep 2020 12:00:00 GMT
Content-Length: 427

{
  "access_token": "79Jdl1lAyNx5ZGyBqEl-N4vefsca9Mwew31gfFQbtSQKjTvzeyQd00",
  "token_type": "bearer",
  "expires_in": 1512000,
  "userName": "مدیر سیستم",
  ".issued": "Wed, 16 Sep 2020 12:00:00 GMT",
  ".expires": "Wed, 14 Oct 2020 12:00:00 GMT"
}
```

پاسخ ارسال شده حاوی مقدار توکن، زمان انقضاء (مقدار تاریخ انقضاء را می توان در تنظیمات کاربر در پنل وب سرویس تغییر داد، مقدار آن در پنل وب سرویس بر اساس ثانیه می باشد) ، تاریخ ایجاد، نام کاربری و نوع توکن می باشد. برای استفاده از این توکن باید مقدار قسمت access_token را به صورتیکه بعدا توضیح داده خواهد شد، در مقدار پارامتر Authorization در header پرس و جوی خود قرار دهید. برای دریافت توکن همچنین می توان از مسیر `http://localhost/pl/token` نیز استفاده کرد. مقدار localhost را با مسیر وب سرویس در سرور خود جایگزین کنید.

نحوه استفاده از توکن

از توکن امنیتی برای اجرای پرس و جوی در وب سرویس مورد استفاده قرار می گیرد. در اینجا از پارامتر Authorization در header پرس و جو، برای ارسال توکن استفاده شده است. از این پارامتر می توان در همه نوع درخواست Post، Get و ... استفاده کرد.

```
GET /db/sample/query1 HTTP/1.1
Host: localhost:80
Connection: keep-alive
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
Authorization: Bearer 79Jdl1lAyNx5ZGyBqEl-N4vefsca9MWew31gfFQbtSQKjTvkeyQd00
Content-Type: application/x-www-form-urlencoded
```

پارامترهای توکن

پارامتر Authorization از قسمت‌های زیر تشکیل شده است:

- Authorization از این پارامتر در header درخواست‌ها برای ارسال مقدار توکن مورد استفاده قرار می‌گیرد.
 - Bearer قبل از مقدار توکن باید نوع آن نیز برای وب سرویس ارسال شود.
 - Token همان مقدار access_token در پاسخ ارسال شده توسط وب سرویس در قسمت قبل می‌باشد.
- مقدار نوع توکن باید حتما با مقدار Bearer قبل از مقدار access_token مشخص شده باشد. در غیر اینصورت پیغام خطا بازگردانده خواهد شد.

Authorization: Bearer 79Jdl1lAyNx5ZGyBqEl-N4vefsca9MWew31gfFQbtSQKjTvkeyQd00

Header پارامتر نوع توکن توکن

معرفی پلاگین

در وب سرویس تدبیر می‌توان با کمک پلاگین‌ها، قابلیت‌های وب سرویس را توسعه داد. پلاگین‌ها، فایل‌های کتابخانه‌ای هستند که می‌توان وارد وب سرویس کرد و متدهای درون آن را فراخوانی کرد. خروجی پلاگین‌ها با توجه به تنظیماتی که برای آن می‌شود از طریق وب به سرویس‌گیرنده منتقل می‌شود.

فایل‌های پلاگین

برای اجرای فایل‌های کاربر، نیاز به ایجاد فایل‌هایی با مشخصات زیر است تا بتوان آنها را در وب سرویس فراخوانی کرد:

- این فایل‌ها از نوع exe و یا dll می‌باشند.
 - در محیط Net Framework نسخه ۳,۵ و یا بعد از آن ایجاد شده باشند.
- فقط کلاس‌هایی قابل فراخوانی هستند که کلاس System.Activities.CodeActivity را پیاده‌سازی کرده باشند. در صورتیکه در فایل پلاگین به فایل دیگری رفرنس داده شده باشد، فایلی که از آن رفرنس گرفته شده نیز باید در کنار فایل پلاگین وارد برنامه شود. فایل رفرنس داده شده نیز باید در محیط Net Framework ایجاد شده باشد. در صورتیکه در فایل پلاگین به کتابخانه‌های Net Framework رفرنس داده شده باشد، نیازی به اضافه کردن این فایل‌ها به برنامه نمی‌باشد.
- می‌توان متادیتای وب سرویس را در فایل پلاگین نگهداری کرد. برای این منظور باید به فایل SPPCPluginExtension.dll در پروژه رفرنس داده شود و Attribute های مورد نیاز پیاده سازی شوند. در غیر اینصورت بعد از وارد کردن فایل باید تنظیمات مورد نیاز انجام شود.

ایجاد پروژه پلاگین

برای ایجاد فایل پلاگین باید در برنامه Visual Studio (ترجیحا آخرین نسخه) ، یک پروژه جدید ایجاد کرد. مشخصات پروژه به ترتیب زیر می‌باشد:

- محیط Net Framework نسخه ۴,۶ یا ۴,۷
- نوع پروژه (.Net Framework Class Library)
- زبان برنامه‌نویسی قابل پیاده‌سازی در محیط Net Framework مانند C#.net, VB.net و یا C++.net

بعد از ایجاد پروژه، در قسمت View\Solution Explorer می‌توان پروژه را مشاهده کرد. در همین قسمت بروی References کلیک کرده و Add Reference را انتخاب کنید. در کادر باز شده گزینه System.Activities را انتخاب کرده

و فرم را تایید کنید. در حال حاضر پروژه آماده است و می‌توان کلاس‌های مورد نیاز را پیاده‌سازی کرد. در صورتیکه در پروژه به غیر از فایل‌های Net Framework. رفرنس می‌دهید، فایل‌های خود را مانند فایل‌های پلاگین وارد وب سرویس کنید.

ایجاد کلاس `CodeActivity`

بعد از آماده کردن پروژه، یک کلاس جدید ایجاد کنید. این کلاس باید از کلاس زیر ارث‌بری داشته باشد.

`System.Activities.CodeActivity<[ResultType]>`

بجای مقدار `ResultType` باید نوع مقداری را که می‌خواهید به وب سرویس منتقل کنید مشخص کنید. مانند `String`، `int`، و یا ... کلاس `CodeActivity` یک کلاس `Abstract` است و باید متد `Execute` آن را در کلاس خود `override` کنید. (مشابه آن در اسلاید بعدی آورده شده است. در اینجا مقدار `ResultType` از نوع `String` می‌باشد.)

نام کلاس را به نام موردنیاز خود تغییر دهید. برای کار کردن با این کلاس، مشابه سایر امکانات قابل پیاده‌سازی در `CodeActivity` می‌توانید کلاس خود را پیاده‌سازی کنید. پارامتر خروجی یا `OutArgument` در وب سرویس پشتیبانی نمی‌شود. برای پارامترهای ورودی می‌توانید از `InArgument` استفاده کنید. هر کلاس فقط می‌تواند یک متد `Execute` و یک خروجی داشته باشد. در نمونه کد زیر کلاس `CodeActivity` پیاده‌سازی شده است. مقدار بازگشتی از نوع `String` است. برای اضافه کردن کد باید قسمت `throw new NotImplementedException();` را حذف کنید و کد خود را جایگزین کنید.

```
using System.Activities;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Sample.Plugin
{
    0 references
    public class Class1 : CodeActivity<string>
    {
        0 references
        protected override string Execute(CodeActivityContext context)
        {
            throw new NotImplementedException();
        }
    }
}
```

پارامترهای ورودی

برای نمونه در کد زیر مقدار `Name` به عنوان پارامتر ورودی دریافت شده و مقدار آن در خروجی به وب سرویس بازگردانده شده است. وب سرویس پارامترهای ورودی را با توجه به متادیتای پارامترهای پرس‌وجو مقداردهی می‌کند.

```

namespace Sample.Plugin
{
    0 references
    public class Class1 : CodeActivity<string>
    {
        1 reference
        public InArgument<string> Name { get; set; }

        0 references
        protected override string Execute(CodeActivityContext context)
        {
            return context.GetValue(Name);
        }
    }
}

```

اضافه کردن متادیتای وب سرویس به فایل پلاگین

برای اضافه کردن متادیتا به فایل پلاگین باید به فایل SPPCPluginExtension.dll در پروژه رفرنس داده شود. این فایل دارای دو Attribute می باشد:

- PluginQueryAttribute برای متادیتای پرس و جو
- PluginQueryParameterAttribute برای متادیتای پارامترهای پرس و جو

پیاده سازی PluginQueryAttribute

از PluginQueryAttribute برای ایجاد سرویس، پرس و جو و مرتبط کردن پرس و جو با گروه‌ها مورد استفاده قرار می‌گیرد. از این Attribute باید قبل از کلاسی که CodeActivity در آن پیاده سازی شده است، استفاده شود. پارامترهای Attribute را باید با توجه به متادیتایی که باید در وب سرویس تعریف شود، مقداردهی شوند. یک فایل پلاگین می‌تواند چندین کلاس CodeActivity و متادیتای مرتبط به آن داشته باشد.

در نمونه کد زیر برای کلاس ۱ متادیتای مرتبط توسط PluginQuery تعریف شده است. در پارامترهای PluginQuery مقدارهای نام سرویس و مسیر سرویس نیز گرفته می‌شود و در صورتیکه در زمان وارد کردن پلاگین، این سرویس‌ها تعریف نشده باشند، یک سطر جدید سرویس در سیستم ایجاد خواهد شد و پرس و جوی مورد نظر به آن مرتبط داده خواهد شد.

```
[SPPC.PluginExtension.PluginQuery(
    name: "Class1",
    path: "class1",
    mime: "application/json",
    serviceName: "service1",
    servicePath: "service1",
    order: 1,
    formatter: SPPC.PluginExtension.PluginFormatter.Json,
    requestType: SPPC.PluginExtension.PluginHttpMethod.Get,
    codePageType: SPPC.PluginExtension.PluginCodePageType.None,
    executionType: SPPC.PluginExtension.PluginExecutionType.Invoke,
    description: "")]
0 references
public class Class1 : CodeActivity<string>
{
    1 reference
    public InArgument<string> Name { get; set; }

    0 references
    protected override string Execute(CodeActivityContext context)
    {
        return context.GetValue(Name);
    }
}
```

پیاده سازی PluginQueryParameterAttribute

از `PluginQueryParameterAttribute` برای تعریف متادیتای پارامترهای پرس‌وجو استفاده می‌شود. در این `Attribute` می‌توان مقدارهای کاربر و یا مقدارهای سیستمی مانند نام کاربری و ... را تعریف و در زمان اجرا وب سرویس با توجه به این داده‌ها، مقدارهای ورودی را به فایل پلاگین ارسال کرد. در متادیتا می‌توان مقدار پارامترهای در سطح سرویس نیز تعریف کرد. اگر پارامتر `ServiceParameterName` مقداردهی شده باشد و در زمان وارد کردن پلاگین این پارامتر وجود نداشته باشد، وب سرویس پارامتر مورد نظر را ایجاد و مرتبط می‌سازد. پارامترهای `Attribute` مشابه پارامترهایی است که در فرم ایجاد پارامترها در وب سرویس استفاده می‌شود.

در نمونه کد زیر برای پارامتر `name` متادیتای مرتبط تعریف شده است. در زمان وارد کردن پلاگین متادیتای تعریف شده نیز به عنوان یک سطر جدید پارامتر تعریف شده و به پرس‌وجوی مورد نظر مرتبط می‌شود. برای سایر پارامترها نیز می‌توان مشابه پارامتر `name` متادیتای مورد نظر را تعریف کرد.


```

public class Class1 : CodeActivity<string>
{
    [SPPC.PluginExtension.PluginQueryParameter(
        order: 1,
        name: "name",
        urlQueryName: "name",
        methodParameterName: "name",
        valueType: "System.String",
        type: SPPC.PluginExtension.PluginParameterType.Custom,
        defaultValue: "",
        notNull: false,
        nonEmpty: false,
        customType: "",
        serviceParameterName: "",
        description: ""
    )]
    1 reference
    public InArgument<string> Name { get; set; }

    0 references
    protected override string Execute(CodeActivityContext context)
    {
        return context.GetValue(Name);
    }
}

```

وارد کردن فایل پلاگین در وب سرویس

برای وارد کردن فایل پلاگین در وب سرویس از منوی [پلاگین / فایل‌های پلاگین / ایجاد] وارد فرم ایجاد پلاگین شده و فایل مورد نظر خود را وارد سیستم کنید. برای دریافت فایل Extension پلاگین برای تعریف متادیتا از طریق فرم [پلاگین / فایل‌های پلاگین] در قسمت پایین فرم [دریافت فایل پلاگین] فایل SPPCPluginExtension.dll را دریافت کنید. در صورت حذف فایل پلاگین، متادیتاهای پرس‌وجو و پارامترهای مرتبط نیز حذف خواهد شد.

نحوه ارسال و دریافت اطلاعات در وب سرویس

معرفی ارسال و دریافت سرویس های پایگاه داده

برای انتقال اطلاعات سرویس های ایجاد شده در وب سرویس پایگاه داده، می توان از امکان ارسال و دریافت اطلاعات در فرم مشاهده سرویس های پایگاه داده استفاده کرد. ارسال اطلاعات و یا export به انتقال یک کپی از اطلاعات به فایل گفته می شود. این اطلاعات وابسته به نسخه وب سرویس می باشد و در نسخه های قبلی قابل استفاده نمی باشد و ممکن است در تمامی نسخه های بعدی کار نکند. (برای اطلاعات بیشتر با کارشناس مرتبط تماس گرفته شود).

وارد کردن اطلاعات و یا import به وارد کردن اطلاعات فایلی که قبلا توسط وب سرویس ایجاد شده است گفته می شود. نسخه فایل باید با نسخه وب سرویس مطابقت داشته باشد. (برای نسخه ۶,۰,۰,۲ این امکان ایجاد شده تا در نسخه های بعدی نیز بتوان از آن استفاده کرد).

ارسال اطلاعات سرویس پایگاه داده

برای ارسال اطلاعات، در صفحه [پایگاه داده / سرویس پرس و جو] و با انتخاب سرویس مورد نظر و دکمه **ارسال سرویس** و بعد از بارگذاری کامل صفحه، بروی صفحه راست کلیک کردن و با انتخاب ذخیره در فایل، اطلاعات ایجاد شده را در یک فایل ذخیره کنید. پسوند فایل مهم نمی باشد و برای انتقال داده ها، نباید در اطلاعات درون فایل تغییری ایجاد شود. این فایل فقط در همان نسخه قابل استفاده می باشد. در نسخه های قدیمی قابل استفاده نمی باشد. در نسخه های جدید ممکن است قابل استفاده نباشد. (برای اطلاعات بیشتر با کارشناس مرتبط تماس گرفته شود).

وارد کردن اطلاعات سرویس پایگاه داده

برای وارد کردن اطلاعات، در صفحه [پایگاه داده / سرویس پرس و جو] و با انتخاب **ورود سرویس** و انتخاب **آخرین نسخه** وارد صفحه ورود سرویس پرس و جو می شوید. برای ورود اطلاعات باید یک سرویس جدید تعریف کرد که شامل مشخصات زیر می باشد:

- نام: نام سرویس می باشد.
- نام مسیر: مسیر اجرای سرویس در URL وب سرویس می باشد.
- فایل سرویس: بعد از انتخاب این گزینه یک پنجره انتخاب فایل باز شده که در آن باید فایلی که قبلا توسط وب سرویس ایجاد شده است را انتخاب کنید.

بعد از تایید فرم، یک سرویس جدید ایجاد شده که شامل کپی پرس و جوها و پارامترهای سرویس ارسال شده می باشد. نام و مسیر وارد شده در این فرم جایگزین نام و مسیر سرویس درون فایل خواهد شد.

وارد کردن اطلاعات سرویس پایگاه داده نسخه ۶,۰,۰,۲

برای وارد کردن اطلاعات، در صفحه [پایگاه داده / سرویس پرس و جو] و با انتخاب ورود سرویس و انتخاب نسخه ۶,۰,۰,۲ وارد صفحه ورود سرویس پرس و جو می شوید. مشابه حالت قبلی اطلاعات فرم را تکمیل کنید. برای انتخاب فایل، انتخاب شده باید حتما قبل از توسط نسخه ۶,۰,۰,۲ ایجاد شده باشد. با تایید فرم، سرویس، پرس و جوها و پارامترهای مرتبط ایجاد خواهد شد. بدلیل تفاوت ساختار URL در نسخه ۶,۰,۰,۲ با نسخه فعلی، نام و مسیر پرس و جوها بعد از ورود، ترکیبی از نام گروه، نام سرویس و .. در نسخه قبلی خواهد بود.

نمونه کد فراخوانی وب سرویس

- فراخوانی پرس و جوها و پلاگین‌ها در وب سایت

یکی از کاربردهای وب سرویس امکان فراخوانی آن توسط جاوا اسکریپت می‌باشد.

```
1 $.ajax("http://webservice:1234/server1/service1/query1?id=12345", {
2     method: 'post',
3     contentType: 'application/json',
4     data: '{ "id": "12345" }',
5     success: function (e) {
6         // Do Something
7     }
8 })
```

همچنین می‌توان با کمک تگ‌های HTML پرس و جوهایی وب سرویس را مقداردهی کرد.

```
1 <html>
2 |   <head></head>
3 <body>
4 |   <form method="POST" action="http://webservice:1234/server1/service1/query1">
5 |       <input name="id" type="text" value="1234" />
6 |       <input name="name" type="text" value="" />
7 |       <button type="submit">تایید</button>
8 |   </form>
9 </body>
10</html>
```

برای نمونه در کد بالا با تایید فرم، مقدار شناسه و نام به عنوان پارامترهای ورودی به وب سرویس ارسال می‌شود.

- فراخوانی سرویس در برنامه‌های اندرویدی

وب سرویس را می‌توان در برنامه‌های اندرویدی و جاوا نیز فراخوانی کرد.

```

1 String url = "http://webservice:1234/server1/service1/query1?id=12345";
2
3 JSONObjectRequest jsonObjectRequest = new JSONObjectRequest
4     (Request.Method.GET, url, null, new Response.Listener<JSONObject>() {
5
6         @Override
7         public void onResponse(JSONObject response) {
8             textView.setText("Response: " + response.toString());
9         }
10    }, new Response.ErrorListener() {
11
12        @Override
13        public void onErrorResponse(VolleyError error) {
14            // TODO: Handle error
15        }
16    }
17    });
18
19 // Access the RequestQueue through your singleton class.
20 MySingleton.getInstance(this).addToRequestQueue(jsonObjectRequest);

```

- فراخوانی سرویس در برنامه‌های دسکتاپ

برای مثال در کد زیر پارامترها بصورت کلاس به وب سرویس ارسال شده است.

```

1  using System.Net.Http;
2  using System.Threading.Tasks;
3
4  namespace Sample
5  {
6      public class QueryParameter
7      {
8          public int Id { get; set; }
9
10         public string Name { get; set; }
11     }
12
13     public class Request
14     {
15         public async Task<string> SendRequest()
16         {
17             QueryParameter parameter = new QueryParameter()
18             {
19                 Id = 1234,
20                 Name = "Name"
21             };
22             System.Net.Http.HttpClient client = new System.Net.Http.HttpClient();
23             var response =
24                 await client.PostAsJsonAsync<QueryParameter>(
25                     "http://webservice:1234/server1/service1/query1",
26                     parameter);
27             if(response.IsSuccessStatusCode)
28             {
29                 return await response.Content.ReadAsStringAsync();
30             }
31             return "";
32         }
33     }
34 }

```