



Graph .NET SDK v5 preview

Highlights

- Generated with new tool called [Kiota](#)
- Enhancements to enrich user experiences across Microsoft Graph
 - Backing store
 - Support \$count endpoints
 - Support for OData Cast endpoints
 - Parameter objects

Developer Experience

V4

```

var requestOptions = new List<Option>
{
    new QueryOption("$select", "subject"),
    new QueryOption("$count", "true")
};

var userEvents = await graphClient.Users["user-id"]
    .Events
    .Request(requestOptions)
    .GetAsync();

```

V5

```
var userEvents = await graphClient.Users["user-id"]
    .Events
    .GetAsync(requestConfig => {
        requestConfig.QueryParameters.Select = new[] { "subject" };
        requestConfig.QueryParameters.Count = true;
    });
```

Backing Store

V4

```
// get the object
var @event = await graphClient.Me.Events["{event-id}"].Request().GetAsync();

// update received object
@event.Recurrence = null;
@event.AdditionalData["recurrence"] = null; // workaround to set null through the additionalData

// update the object
await graphClient.Me.Events["{event-id}"].Request().UpdateAsync(@event);
```

V5

```
// get the object
var @event = await graphClient.Me.Events["{event-id}"].GetAsync();

// the backing store will keep track that the property change and only send the updated properties.
@event.Recurrence = null;

// update the object
await graphClient.Me.Events["{event-id}"].PatchAsync(@event);
```

Parameter objects

V4

```
·var·message·=·new·Message
·{
·····Subject·=·"Meet·for·lunch?",
·····Body·=·new·ItemBody[...],
·····ToRecipients·=·new·List<Recipient>()[...]
·};
```

```
·await·graphClient.Me.SendMail(message,·SaveToSendItems:·true).Request().PostAsync();
```

V5

```
·var·requestBody·=·new·SendMailPostRequestBody
·{
·····Message·=·new·Message
·····{
·········Subject·=·"Meet·for·lunch?",
·········Body·=·new·ItemBody[...],
·········ToRecipients·=·new·List<Recipient>()[...]
·····},
·····SaveToSendItems·=·true
·};
```

```
·await·graphClient.Me.SendMail.PostAsync(requestBody);
```

Support for \$count segments

```
//·Call·https://graph.microsoft.com/v1.0/users/\$count  
var·count·=·await·graphClient.Users.Count.GetAsync(requestConfig·=>·{  
·.....requestConfig.Headers.Add("ConsistencyLevel",·"eventual");  
·.....});
```

Support for OData Cast segments

```
//·call·GET·https://graph.microsoft.com/v1.0/groups/group-id/members/microsoft.graph.user  
var·usersInGroup·=·await·graphClient.Groups["group-id"].Members.User.GetAsync();
```

aka.ms/graph/sdk/dotnet-preview