



Armen Melkumyan • 1st
Technical / Solutions Architect
10mo •

...

From .Net C# Interview Questions

Exploring Task.Factory in C#:

Task.Factory: Part of the Task Parallel Library (TPL)

Provides methods for creating and starting Task instances.

More configurable than basic Task constructors or [Task.Run](#).

Key Features of Task.Factory:

Task Creation: Various methods for tailored task configurations.

Custom Task Schedulers: Specify a TaskScheduler for bespoke scheduling.

Task Creation Options: Supports diverse TaskCreationOptions for enhanced control.

Continuation Tasks: Easy creation of tasks that start after another task completes.

Usage Examples:

Opt for LongRunning tasks for dedicated threading.

Define a custom TaskScheduler for precise execution control.

Utilize continuation tasks for sequential task flows.

[#CSharpProgramming](#) [#DotNetDevelopers](#) [#Task](#) [#TaskFactory](#) [#TPL](#)

[#ParallelComputing](#) [#CodingInterviews](#)

```
1 // Basic Example
2 Task.Factory.StartNew(() =>
3 {
4     // Your code here
5     Console.WriteLine("Task running");
6 }, CancellationToken.None, TaskCreationOptions.None, TaskScheduler.Default);
7
8
9 // LongRunning example
10
11 Task longRunningTask = Task.Factory.StartNew(() =>
12 {
13     // Long running operation
14 }, TaskCreationOptions.LongRunning);
15
16 // TaskScheduler example
17 TaskScheduler customScheduler = /* your custom scheduler */;
18 Task customScheduledTask = Task.Factory.StartNew(() =>
19 {
20     // Code to execute
21 }, CancellationToken.None, TaskCreationOptions.None, customScheduler);
```

Armen Melkumyan

