# Cemit.Awaiter

## 返回AsyncAction以支持await关键字

1. **class** AsyncOperator
2. {
3. **public** **static** AsyncAction<**int**> Run()
4. {
5. AsyncAction<**int**> action = **new** AsyncAction<**int**>();
6. **return** action;
7. }
8. }
10. **class** Program
11. {
12. **static** **void** Main(**string**[] args)
13. {
14. Console.WriteLine("start");
15. AsyncLog();
16. }
18. **private** **static** async **void** AsyncLog()
19. {
20. **int** i = await AsyncOperator.Run();
21. }
22. }

## 调用AsyncAction<T>. ReportResult(T)以返回结果，并以一个新的线程调用await关键字之后的代码

1. **class** DelayRandom
2. {
3. **static** **readonly** Random random;
5. **static** DelayRandom()
6. {
7. random = **new** Random((**int**)DateTime.Now.ToFileTime());
8. }
10. **public** **static** AsyncAction<**int**> Next(**int** delay, **int** min, **int** max)
11. {
12. AsyncAction<**int**> action = **new** AsyncAction<**int**>();
14. //模拟延迟（耗时）操作
15. **new** Thread(() =>
16. {
17. //延迟一段时间
18. Thread.Sleep(delay);
19. //调用结果，并运行调用者await之后的代码
20. action.ReportResult(random.Next(min, max));
22. }).Start();
24. **return** action;
25. }
26. }
28. **class** Program
29. {
30. **static** **void** Main(**string**[] args)
31. {
32. Console.WriteLine("start");
33. LogRandom();
34. }
36. **private** **static** async **void** LogRandom()
37. {
38. **int** i = await DelayRandom.Next(3000, 1, 100);
39. //在ReportResult之后执行
40. Console.WriteLine(i);
41. }
42. }

## 调用AsyncAction<>. ThrowException(Exception) 抛出一个错误

1. **if** (min > max)
2. {
3. action.ThrowException(**new** Exception("min不能大于max"));
4. **return**;
5. }