USING BACKGROUND PROCESSING TO BUILD SCALABLE APPLICATIONS WITH HANGFIRE

Lab 5

Goal

Hangfire works by dynamically invoking jobs at runtime. In this lab we will take a deeper look at how Hangfire works by updating our House building job so that our Lemmings are now buildings houses with friends.

1. Open the BuildHouseJob class inside of Core.Jobs
2. Implement a BuildHouseWithFriend method by copying the following lines into the class:

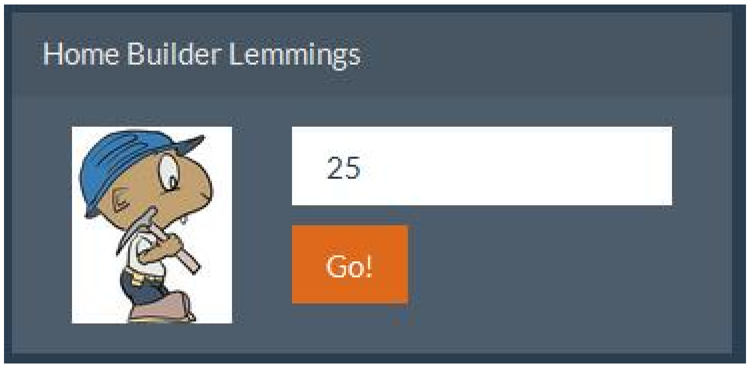
```
public void BuildHouseWithFriend()
        {
            Logger.Info($"Building a house with my friend!");
            UpdateDashboard<BuildHouse>();
        }
```

3. Open the HomeController and update the DoBuildHouseJob Action to call the BuildHouseWithFriend method:

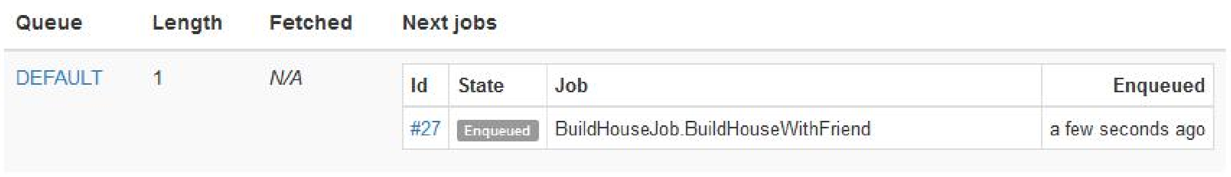
```
public ActionResult DoBuildHouseJob(int? quantity)
        {
            RunAsync<BuildHouseJob>(j => j.BuildHouseWithFriend(), quantity);

            return RedirectToAction("Index");
        }
```

4. Now run the application by pressing F5 and queue up some BuildHouse Jobs (which will have your Lemmings now building houses with a friend).



5. Navigate to http://localhost:13161/hangfire/jobs/enqueued and verify that you have some BuildHouseWithFriendJobs that have been queued and/or completed.



Queue	Length	Fetched	Next jobs
DEFAULT	1	N/A	

Id	State	Job	Enqueued
#27	Enqueued	BuildHouseJob.BuildHouseWithFriend	a few seconds ago

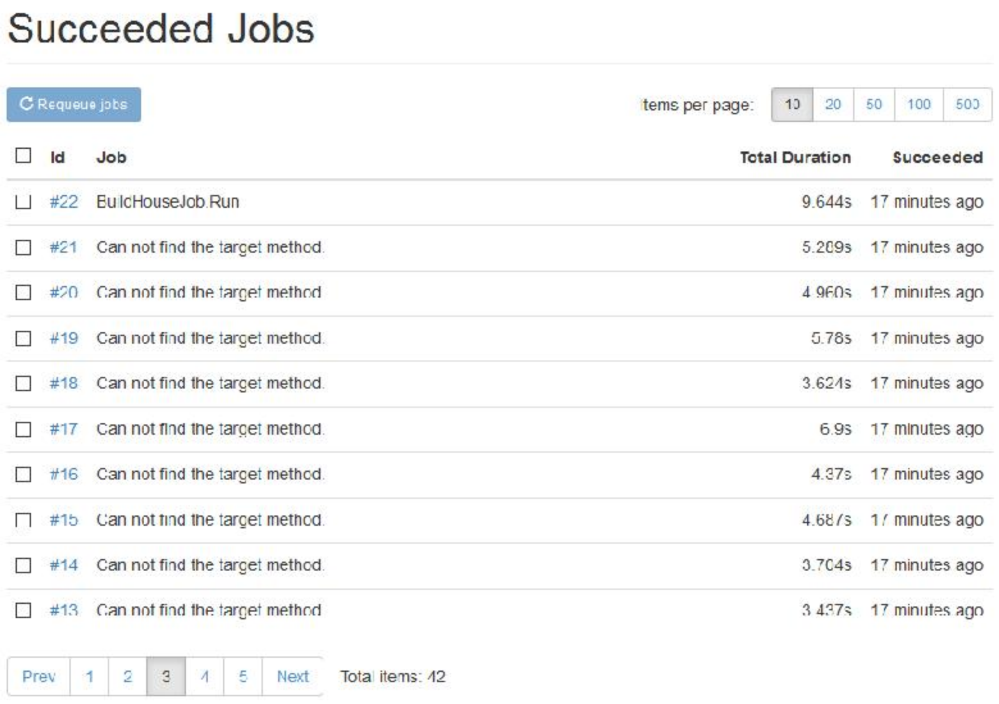
6. Next, we are going to update our BuildHouseWithFriendJob and add a parameter to it. Open the BuildHouseJob class and update the method:

```
public void BuildHouseWithFriend(string friendName)
        {
            Logger.Info($"Building a house with my friend {friendName}!");
            UpdateDashboard<BuildHouse>();
        }
```

7. Open the HomeController and update the call to the BuildHouseWithFriendJob with your friend's name as a parameter:

```
RunAsync<BuildHouseJob>(j => j.BuildHouseWithFriend("David"), quantity);
```

8. Start your project again by pressing F5 and navigate to your succeeded jobs: http://localhost:13161/hangfire/jobs/succeeded
9. Look for your BuildHouseWithFriendJob by paging through the succeeded job list. Notice that it does not show up but you have a strange job name there now (can not find the target method).



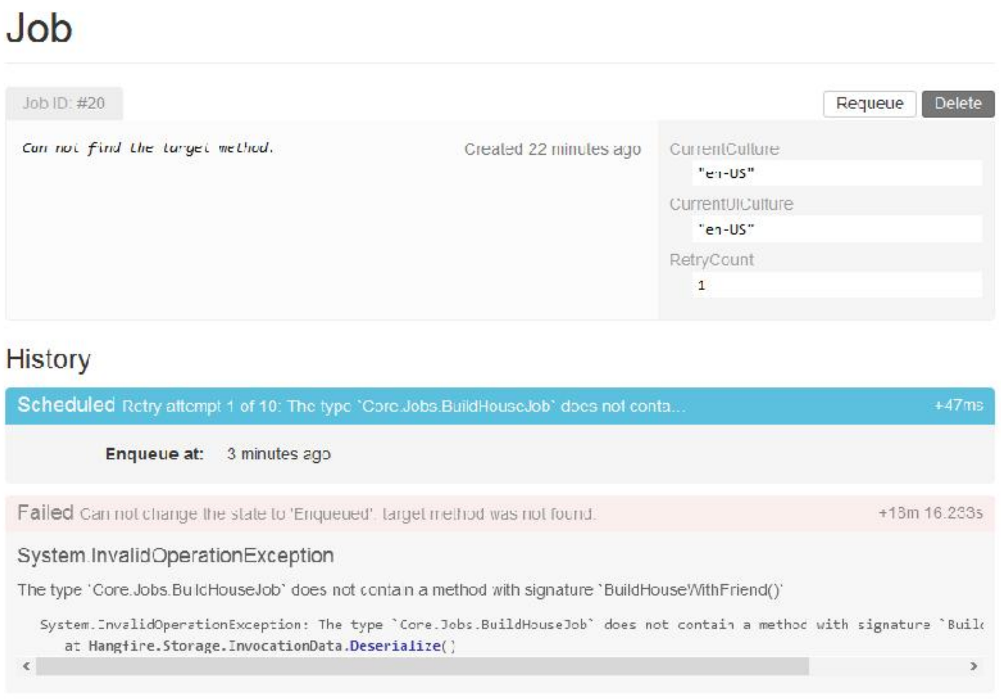
Succeeded Jobs

☐	Id	Job	Total Duration	Succeeded
☐	#22	BuildHouseJob.Run	9.644s	17 minutes ago
☐	#21	Can not find the target method.	5.209s	17 minutes ago
☐	#20	Can not find the target method.	4.960s	17 minutes ago
☐	#19	Can not find the target method.	5.78s	17 minutes ago
☐	#18	Can not find the target method.	3.624s	17 minutes ago
☐	#17	Can not find the target method.	6.9s	17 minutes ago
☐	#16	Can not find the target method.	4.37s	17 minutes ago
☐	#15	Can not find the target method.	4.687s	17 minutes ago
☐	#14	Can not find the target method.	3.704s	17 minutes ago
☐	#13	Can not find the target method.	3.437s	17 minutes ago

Total items: 42

This message is showing because Hangfire can no longer find the method signature in your code base.

10. Click on the Id link of one of the jobs then click the requeue button. After a few moments refresh the page and you will notice that you have an exception that was thrown when trying to run the job (because Hangfire cannot find the method signature).



```
Can not find the target method.
```

Created 22 minutes ago

CurrentCulture
"en-US"
CurrentUICulture
"en-US"
RetryCount
1

History

Scheduled Retry attempt 1 of 10: The type `Core.Jobs.BuildHouseJob` does not conta... +47ms

Enqueue at: 3 minutes ago

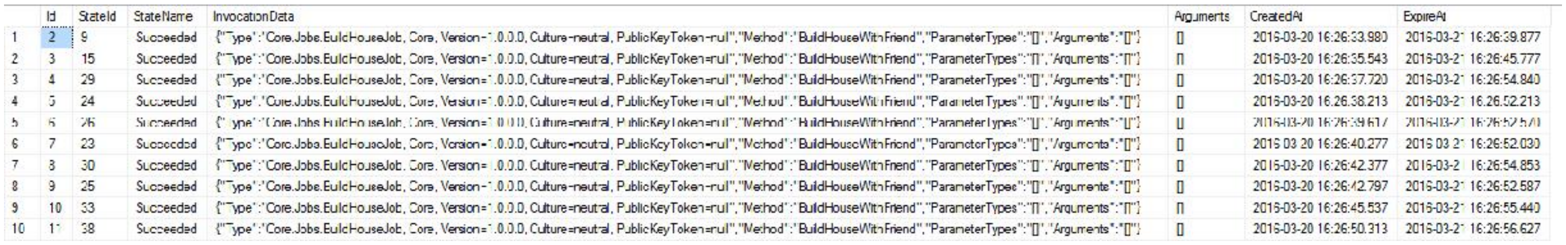
Failed Can not change the state to 'Enqueued': target method was not found. +13m 16.233s

System.InvalidOperationException

The type `Core.Jobs.BuildHouseJob` does not contain a method with signature `BuildHouseWithFriend()`

```
System.InvalidOperationException: The type `Core.Jobs.BuildHouseJob` does not contain a method with signature `Buil
   at Hangfire.Storage.InvocationData.Deserialize()
```

Let's take a look at how Hangfire is storing information about the jobs in the database.



	Id	StateId	StateName	InvocationData	Arguments	CreatedAt	ExpireAt
1	2	9	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:33.980	2016-03-21 16:26:39.877
2	3	15	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:35.543	2016-03-21 16:26:45.777
3	4	29	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:37.720	2016-03-21 16:26:54.840
4	5	24	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:38.213	2016-03-21 16:26:52.213
5	6	26	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:39.617	2016-03-21 16:26:52.570
6	7	23	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:40.277	2016-03-21 16:26:52.030
7	8	30	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:42.377	2016-03-21 16:26:54.853
8	9	25	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:42.797	2016-03-21 16:26:52.587
9	10	33	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:45.537	2016-03-21 16:26:55.440
10	11	38	Succeeded	{"Type":"Core.Jobs.BuildHouseJob, Core, Version=1.0.0.0, Culture=neutral, PublicKeyToken=null","Method":"BuildHouseWithFriend","ParameterTypes":"[]","Arguments":"[]"}	[]	2016-03-20 16:26:50.313	2016-03-21 16:26:56.627

If you want to view the data in your database you can run the following query:

```sql
USE LemmingSchedulingSystem
SELECT * FROM HangFire.Job
WHERE InvocationData LIKE '%BuildHouseWith%'
```

Notice the "Type" property in the data. Hangfire uses this to dynamically invoke your job using Activator.CreateInstance (https://msdn.microsoft.com/en-us/library/system.activator.createinstance(v=vs.110).aspx)

When you deploy an update to your application, it is possible that there are jobs waiting in your queue that have not yet been processed. Therefore, when changing the signatures of your method you need to ensure that you are still processing the "old" version of the jobs that are in the queue. In later versions of our project code we can remove the old method (once we know they are all out of the queue). Let's take a look to see how we can fix this.

11. Open the BuildHouseJob class and add a new method to this class:

```
public void BuildHouseWithFriendWithName(string friendName)
        {
            Logger.Info($"Building a house with my friend {friendName}!");
            UpdateDashboard<BuildHouse>();
        }
```

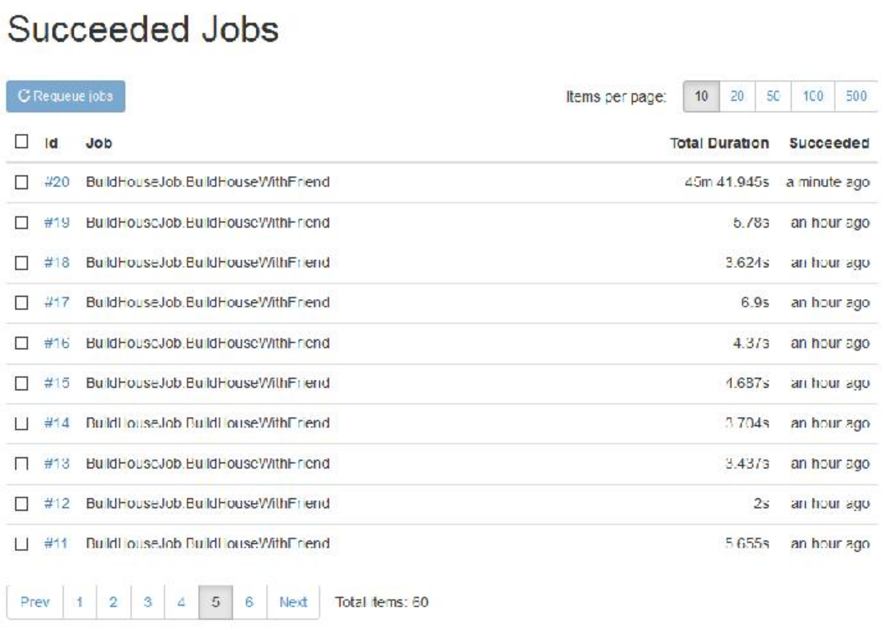
12. Update the old method to call the new method and send in a default parameter:

```
public void BuildHouseWithFriend()
        {
            BuildHouseWithFriendWithName("Default");
        }
```

13. Update the method call in the HomeController to call the newly made method:

```
RunAsync<BuildHouseJob>(j => j.BuildHouseWithFriendWithName("David"), quantity);
```

14. Run the project by pressing F5.
15. Navigate back to http://localhost:13161/hangfire/jobs/succeeded and ensure that you no longer see the "Can not find the target method" message.



Succeeded Jobs

Id	Job	Total Duration	Succeeded
#20	BuildHouseJob.BuildHouseWithFriend	45m 41.945s	a minute ago
#19	BuildHouseJob.BuildHouseWithFriend	5.78s	an hour ago
#18	BuildHouseJob.BuildHouseWithFriend	3.624s	an hour ago
#17	BuildHouseJob.BuildHouseWithFriend	6.9s	an hour ago
#16	BuildHouseJob.BuildHouseWithFriend	4.37s	an hour ago
#15	BuildHouseJob.BuildHouseWithFriend	4.687s	an hour ago
#14	BuildHouseJob.BuildHouseWithFriend	3.704s	an hour ago
#13	BuildHouseJob.BuildHouseWithFriend	3.437s	an hour ago
#12	BuildHouseJob.BuildHouseWithFriend	2s	an hour ago
#11	BuildHouseJob.BuildHouseWithFriend	5.655s	an hour ago

Total items: 60

16. Requeue a BuildHouseWithFriend job and notice that it now succeeds.
17. Lastly, go back to the Lemming Dashboard (http://localhost:13161/) and notice that the new jobs are also succeeding.

This completes Lab 05.