# 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>();

}

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

public ActionResult DoBuildHouseJob(int? quantity)

{

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

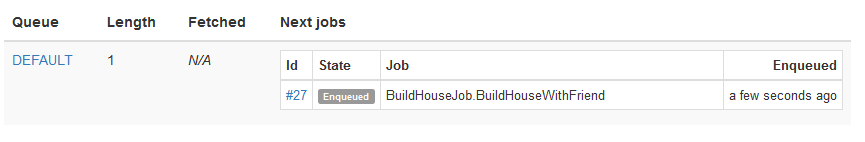
return RedirectToAction("Index");

}

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



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



1. 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}!");

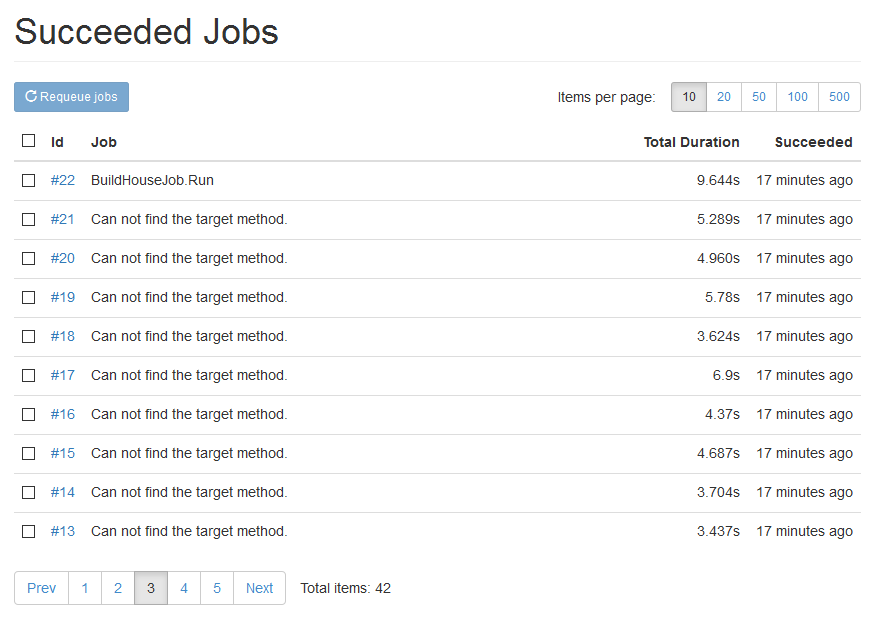
UpdateDashboard<BuildHouse>();

}

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

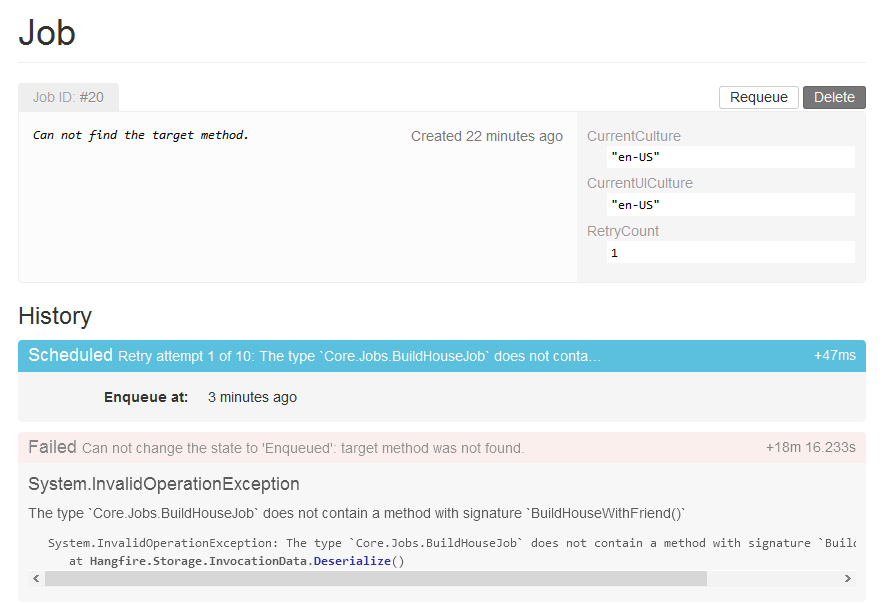
RunAsync<BuildHouseJob>(j => j.BuildHouseWithFriend("David"), quantity);

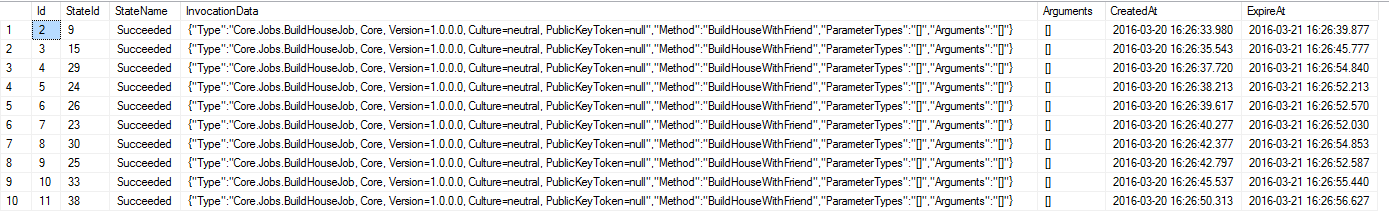
1. Start your project again by pressing F5 and navigate to your succeeded jobs: <http://localhost:13161/hangfire/jobs/succeeded>
2. Look for your BuildHouseWithFriendJob by paging through the succeeded job list. Notice that it does not show up but you have a strange job name there now (can not find the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



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

1. 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).



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

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

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.

1. 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>();

}

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

public void BuildHouseWithFriend()

{

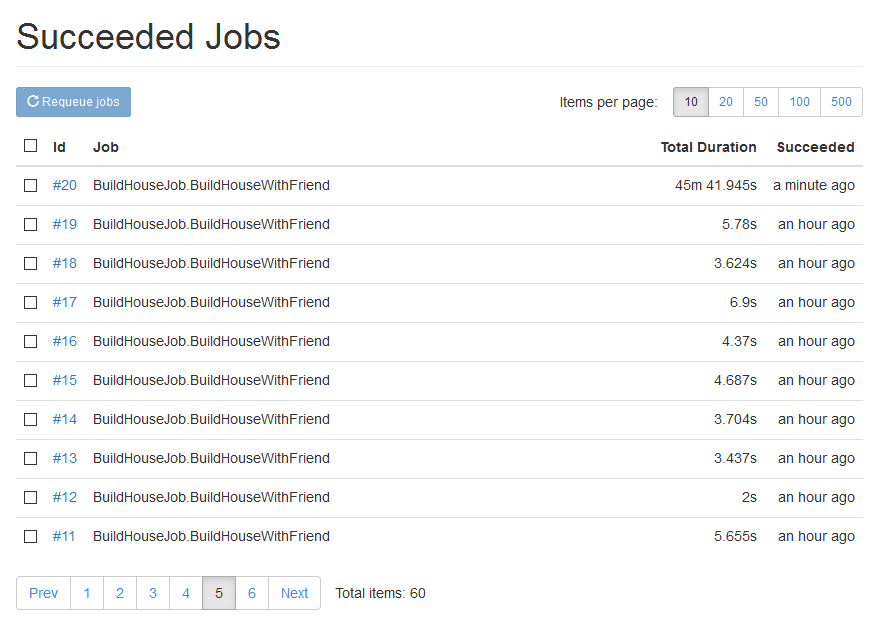
BuildHouseWithFriendWithName("Default");

}

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

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

1. Run the project by pressing F5.
2. Navigate back to <http://localhost:13161/hangfire/jobs/succeeded> and ensure that you no longer see the “Can not find the target method” message.



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

This completes Lab 05.