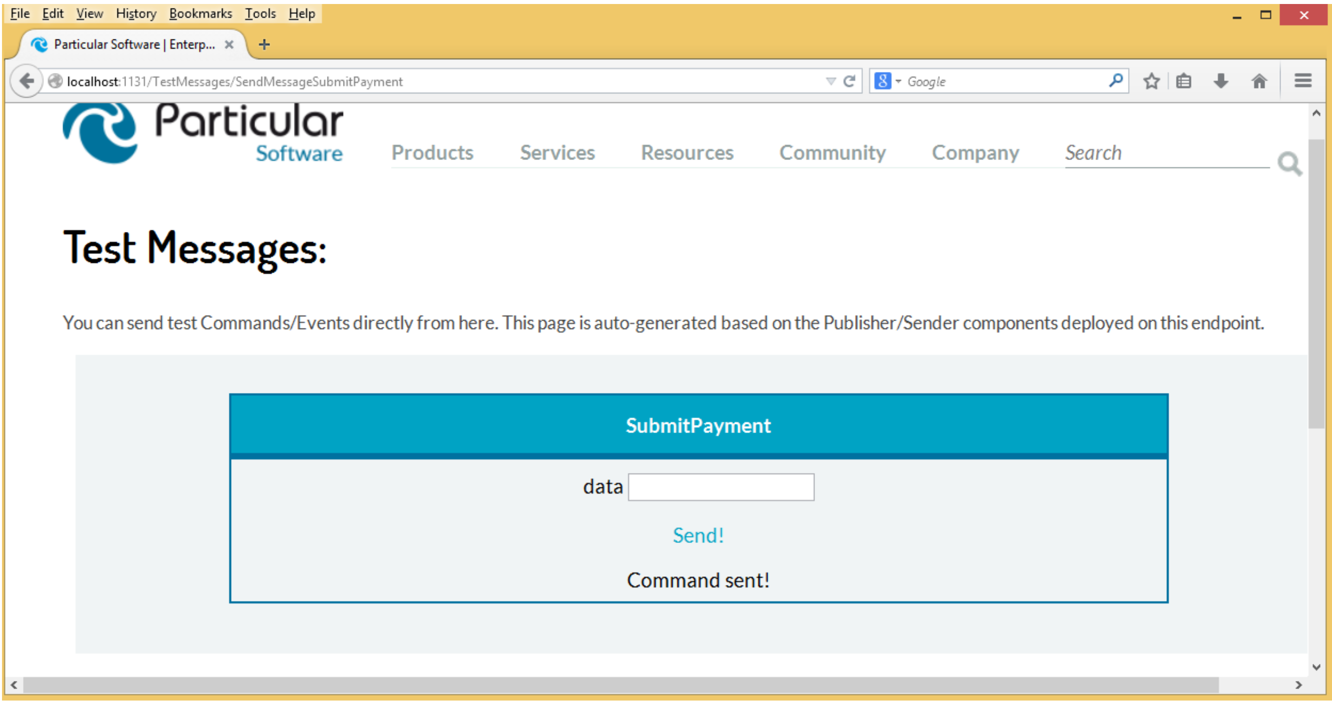
Chapter 3, *Particular Service Platform*, is an overview of the Particular.net associated tools for NServiceBus. We will discuss building Sagas through the ServiceMatrix tool which is a Visual Studio extension tool for visually designing NSB endpoints, messages and services. The other tools that we will discuss at length as they apply to Sagas are the ServicePusle for monitoring the endpoint availability during production, and ServiceInsight for taking a deep dive into the functionality and properties of endpoints, services and messages as they execute.

The Source code in this section:

In this section the NServiceBus Service platform must be installed to include ServiceMatrix, ServicePulse and ServiceInsight.

In this section, we will be using the **PaymentEngine-Start** solution created with the ServiceMatrix, so ServiceMatrix must be installed. This will be the ServiceMatrix walkthrough. This will be a request-repsonse solution built with ServiceMatrix. The outcome of a sent payment will look as follows in a Payment:



Test Messages:

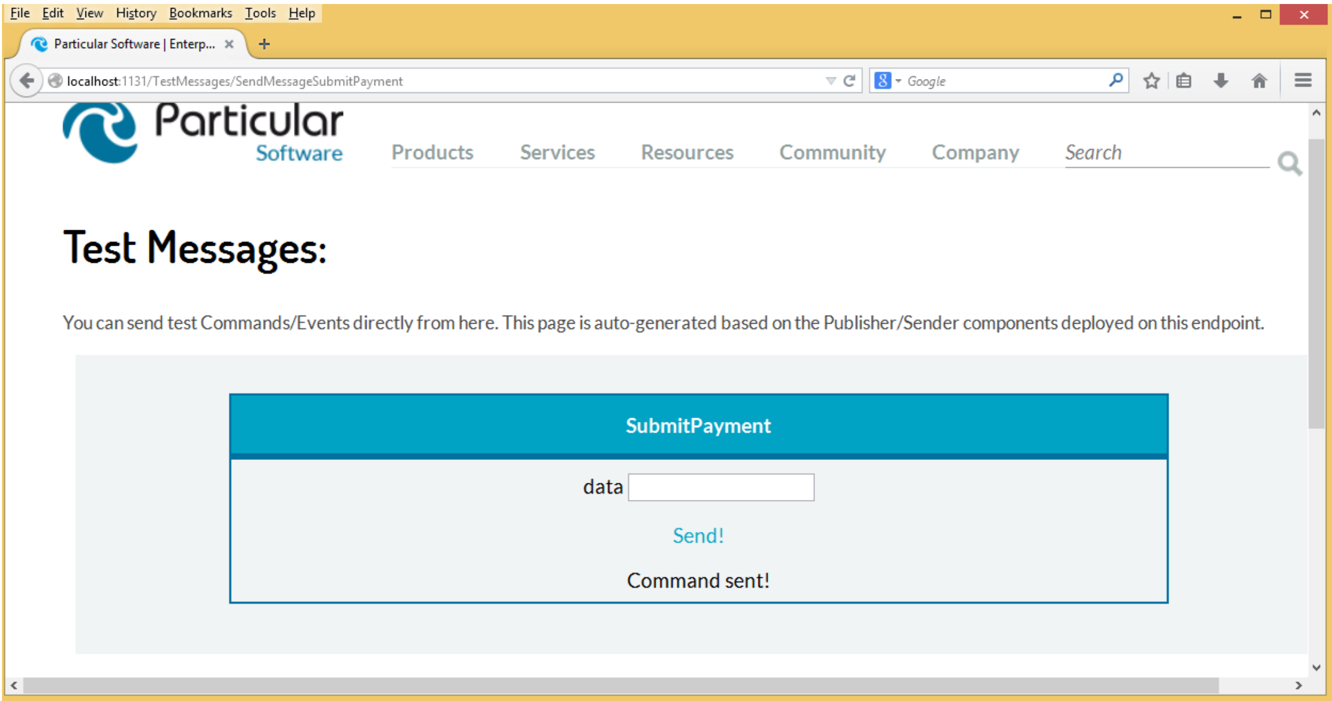
You can send test Commands/Events directly from here. This page is auto-generated based on the Publisher/Sender components deployed on this endpoint.

SubmitPayment

data

Command sent!

In this section, we will be using the **PaymentEngine-Saga** solution created with the ServiceMatrix, so ServiceMatrix must be installed. This will be the ServiceMatrix walkthrough. This will be a publish-subscribe and Saga solution built using ServiceMatirx. The outcome of a sent payment will look like::


File Edit View History Bookmarks Tools Help
Particular Software | Enterp...
localhost:1131/TestMessages/SendMessageSubmitPayment
Google
Particular Software
Products
Services
Resources
Community
Company
Search

Test Messages:

You can send test Commands/Events directly from here. This page is auto-generated based on the Publisher/Sender components deployed on this endpoint.

SubmitPayment

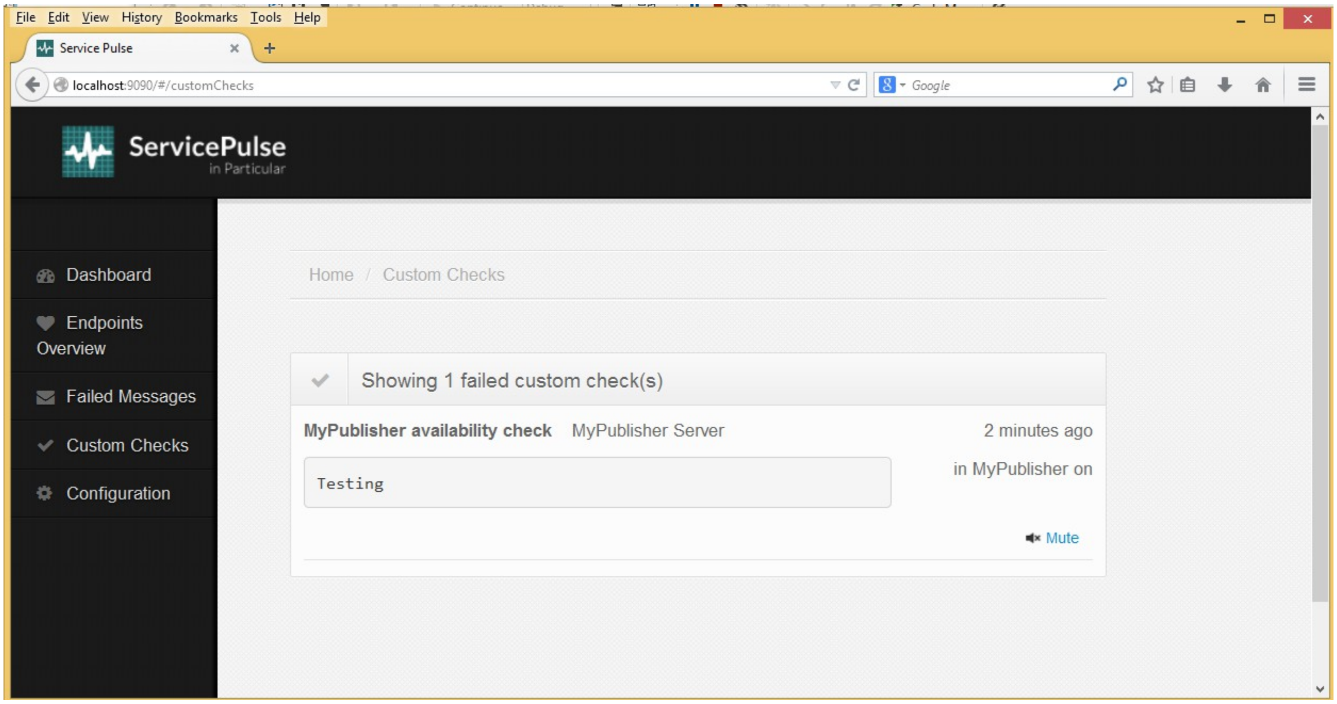
data

Send!

Command sent!

In this section, we will be using the **PubSub--ReportFailure** solution, the MyPublisher project reports a failure check that will be reported in the ServicePulse. This shows Custom Checks.

A failure check sent to ServicePulse will look like:



Showing 1 failed custom check(s)

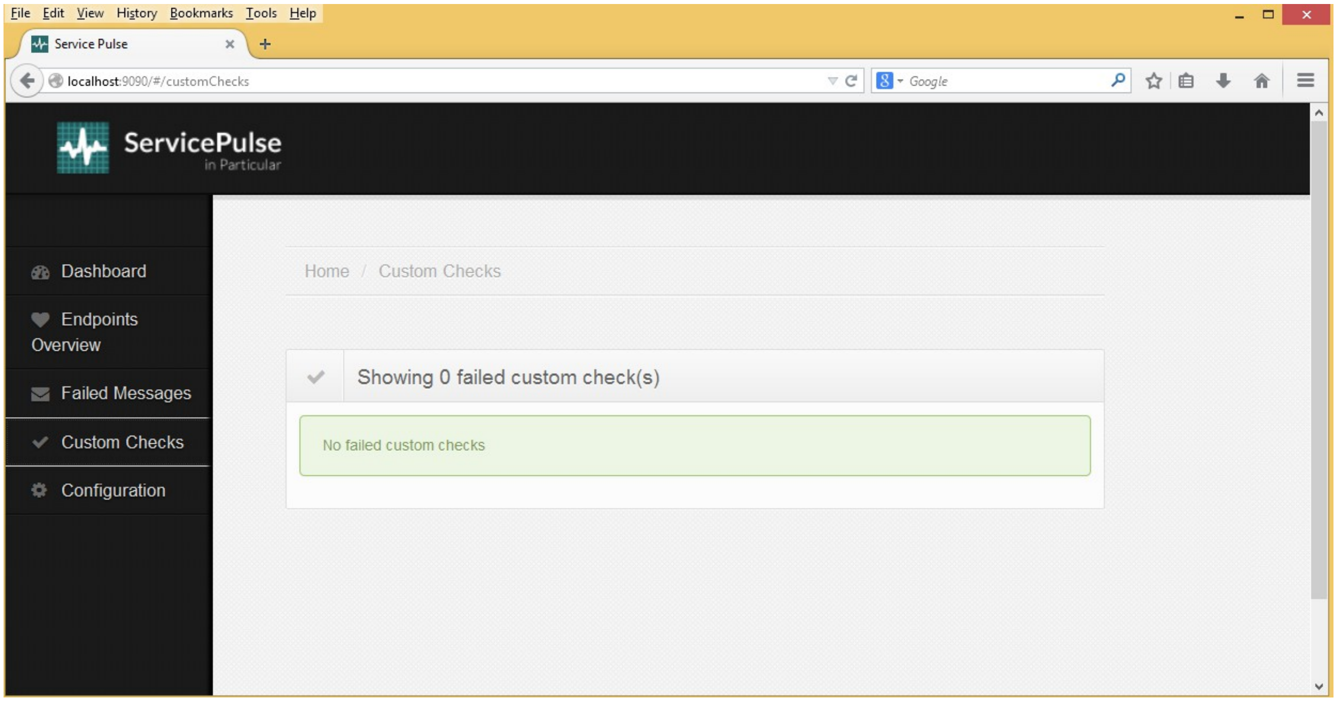
MyPublisher availability check MyPublisher Server 2 minutes ago

```
Testing
```

in MyPublisher on

In this section, we will be using the **PubSub--ReportPass** solution, the MyPublisher project reports a pass check that will be reported in the ServicePulse. This shows Custom Checks.

A report pass in ServicePulse will look like:



Showing 0 failed custom check(s)

No failed custom checks