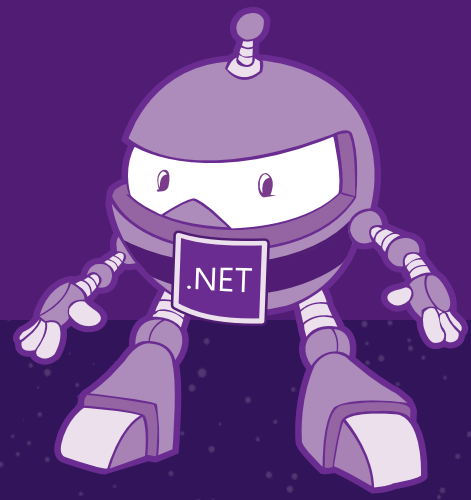
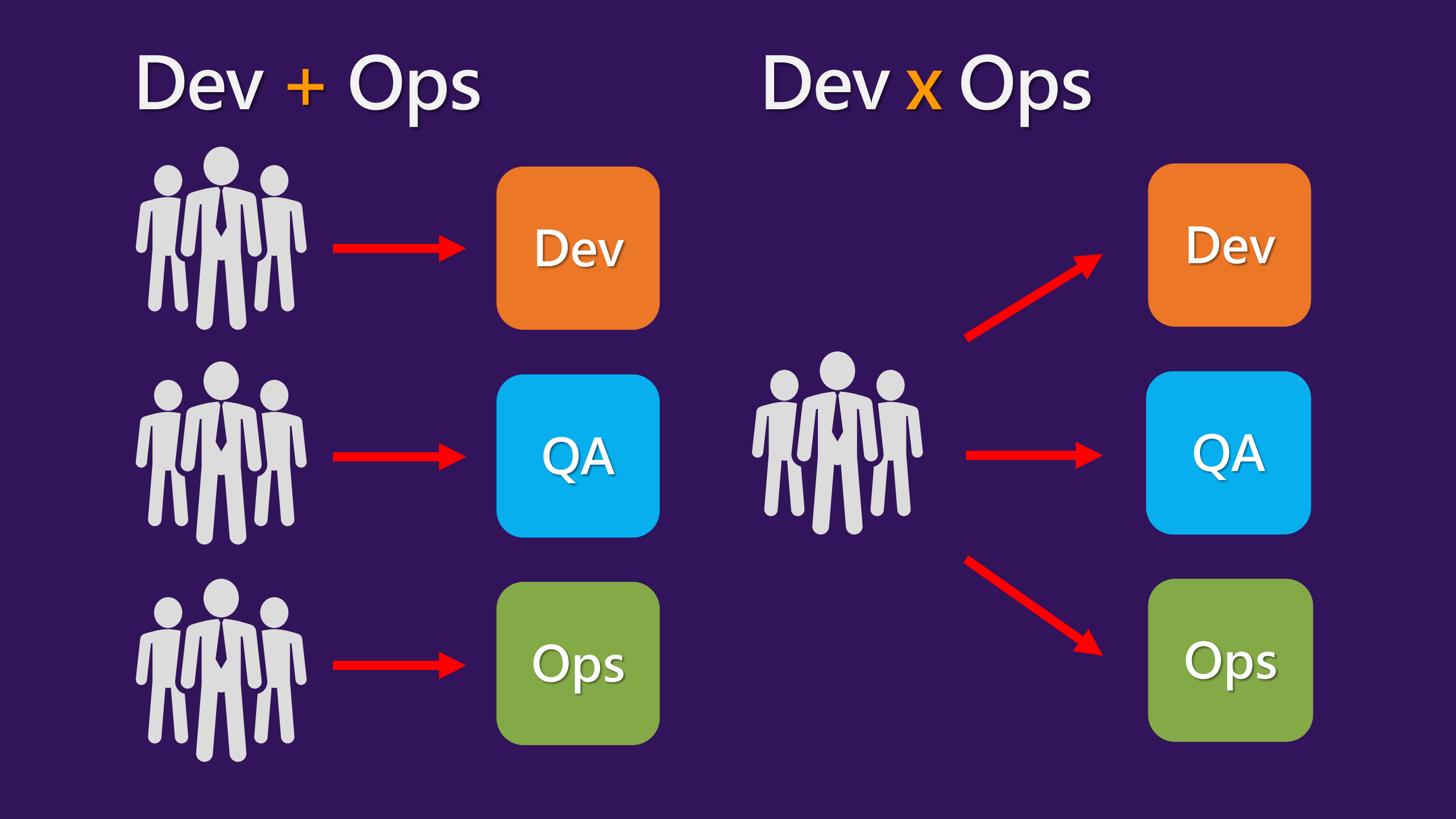

.NET Conf 2019
Discover the world of .NET
dotnet new
.NET
https://mvc.tw
MVC

讓Exceptionless存管你的Log

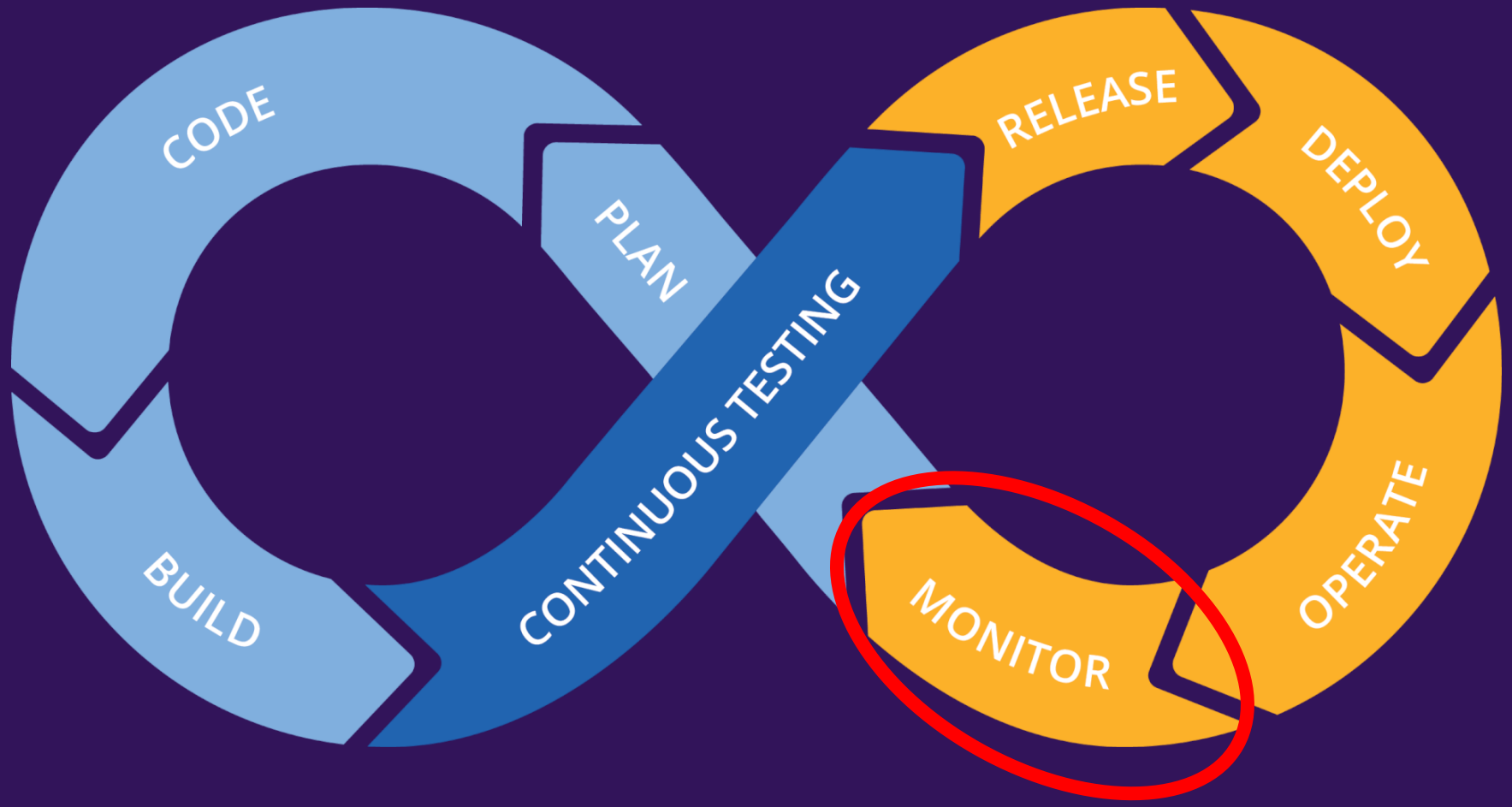
Roberson Liou



tw MVC
https://mvc.tw


Dev + Ops
Dev
QA
Ops
Dev x Ops
Dev
QA
Ops

DevOps Circle

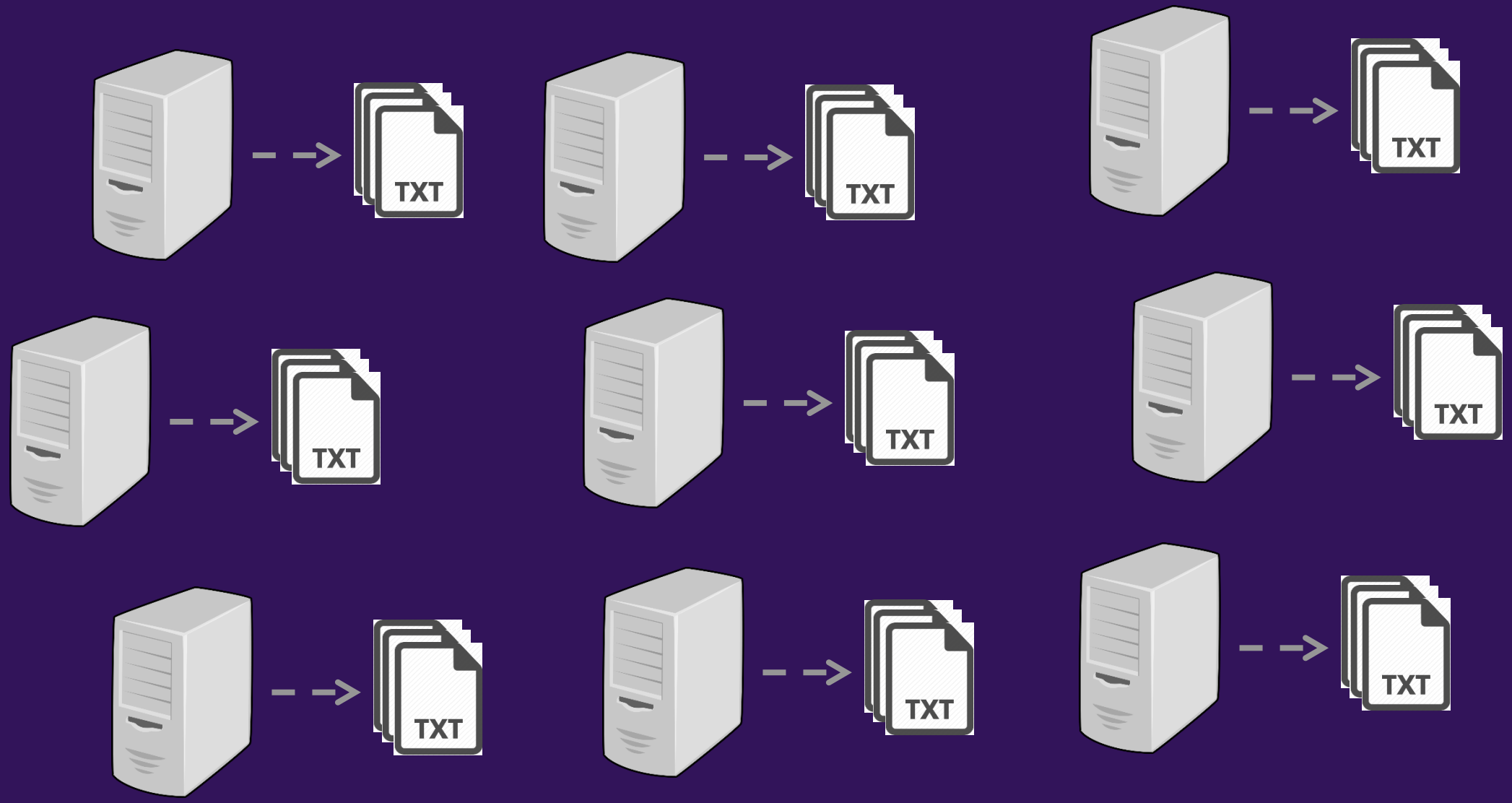


Why we started Exceptionless



一切都從混亂開始...

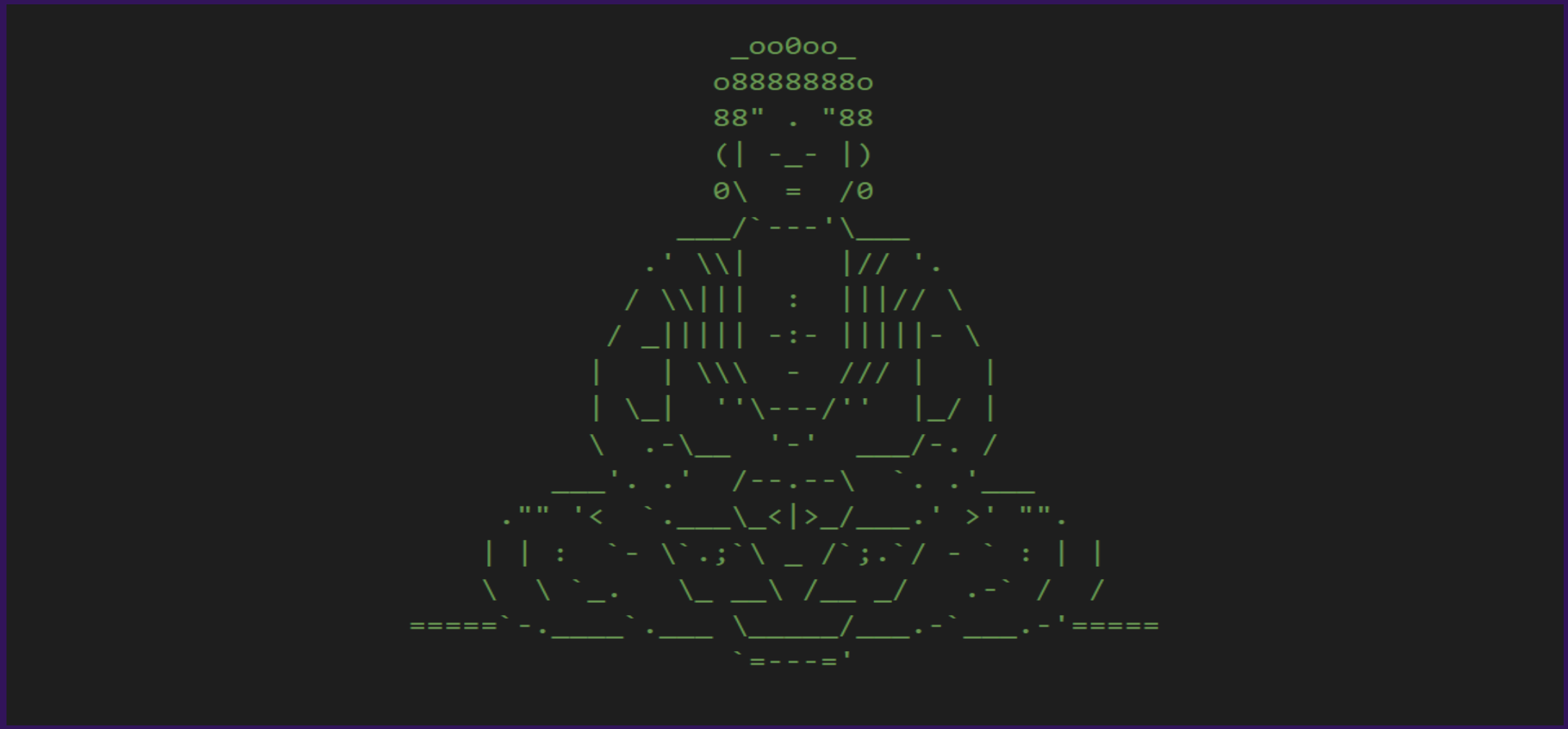
Problems with Logging



過度繁雜導致Log查找不易



有寫有保庇，沒寫出代誌



.Net + Log = ?

- Request/Response
- Dependency Services
- Exceptions
- Page views and load performance
- Ajax Calls





Application Insights

- User Session Counts
- Performance counters
- Host diagnostics
- Diagnostic trace logs
- Custom events and metrics



Exceptionless

Exceptionless簡介

- Open Source
- 整合Exception、Log等事件
- 支援多種.Net開發框架
- 提供雲端及自架服務(IIS、Docker)
- 友善的使用者介面
- 提供整合的訊息平臺入口
- 提供高客製化的擴充彈性



Blake Niemyjski niemyjski

A full time open source software architect, Microsoft MVP, beekeeper and private pilot.

Suamico, Wisconsin

Member of Exceptionless, Foundatio, and 2 more

功能簡介

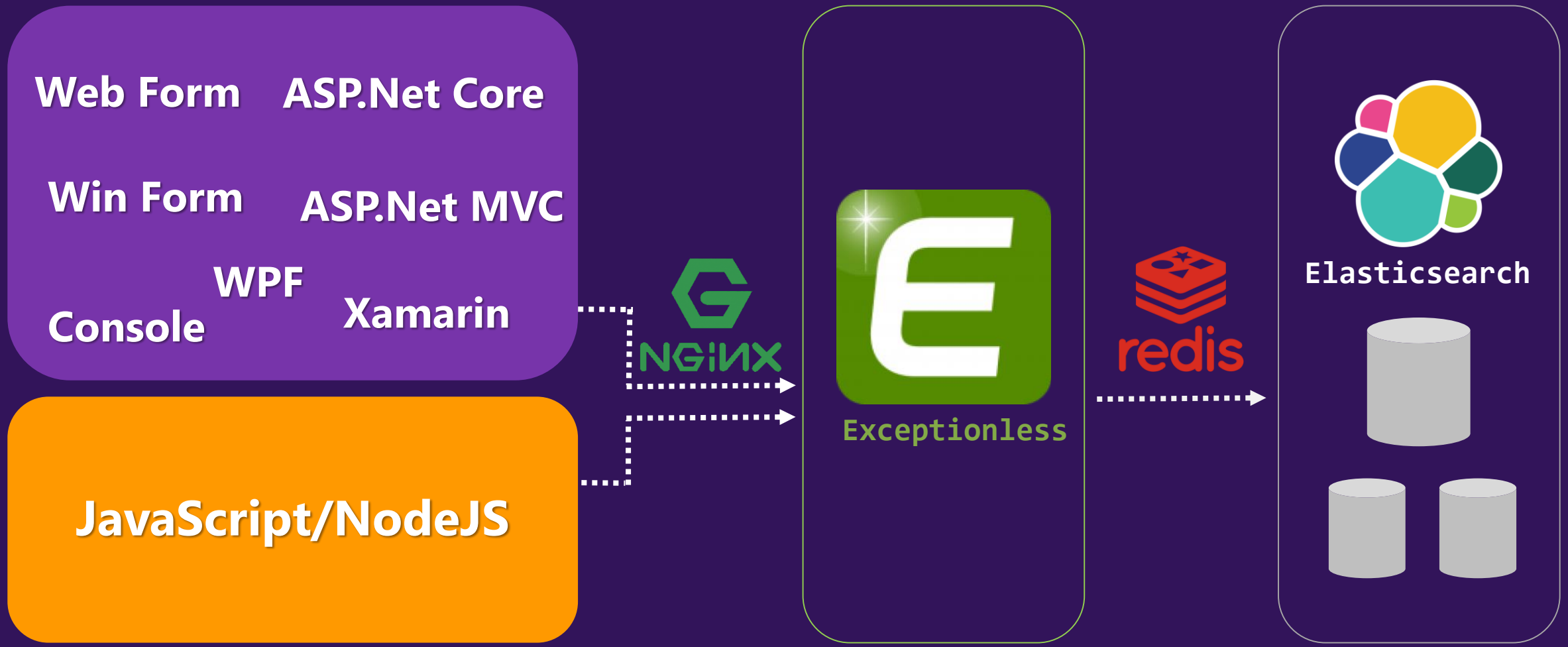
- **Exceptions**
  (例外記錄)
- **Logs**
  (日誌記載)
- **Broken Links**
  (失效連結)
- **Feature Usages**
  (功能統計)



Exceptionless

- **User Sessions**
  (使用紀錄)
- **Integrations**
  (外部整合)
- **Data Exclusions**
  (資料過濾)
- **Self Hosted**
  (自架服務)

系統架構


Web Form
ASP.Net Core
Win Form
ASP.Net MVC
WPF
Console
Xamarin
JavaScript/NodeJS
NGINX
Exceptionless
redis
Elasticsearch



Exceptionless Server篇

雲端服務(1)-簡介

- 可快速測試相關API服務
- 無須額外架設服務
- 無須維護及配置機器
- 無須升級即可享有最新服務

雲端服務(2)-計價方案

方案	計價(美金/月)	專案/帳戶數	事件保留天數	事件次數/月	進階功能
Free	$0	1/1	3	3000	N
Small	$15	5/10	30	15000	Y
Medium	$49	15/25	90	75000	Y
Large	$99	無限制	180	250,000	Y
Extra Large	$199	無限制	180	1,000,000	Y
Enterprise	$499	無限制	180	3,000,000	Y

※進階功能：**Searching**、**Notification**、**WebHook**、**Slack Integration**、**Zapier Integration**

自架服務(Self Hosted)

- 下載最新release版本
- 調整docker-compose.yml
- 官方建議Elasticsearch數 : 3 nodes, 2 masters
- 執行指令
  - ✓docker-compose up
  - ✓docker-compose up –d (背景執行)
  - ✓docker-compose stop



帳戶管理(1)

- 一個Exceptionless可以新增多個帳戶
- 一個帳戶可以新增多個組織
- 一個組織可以新增多個專案
- 帳戶之間可以設定**共同管理**組織

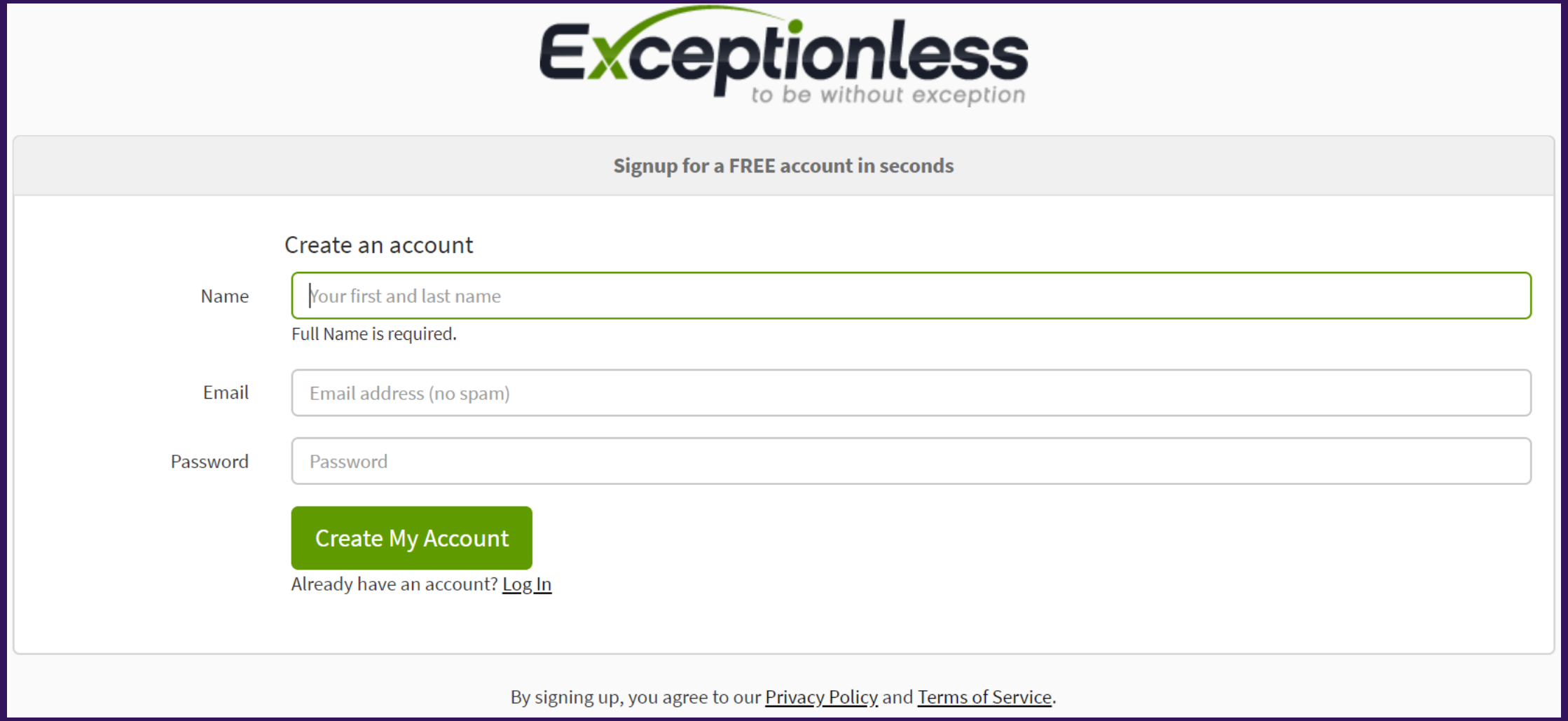
帳戶管理(2)



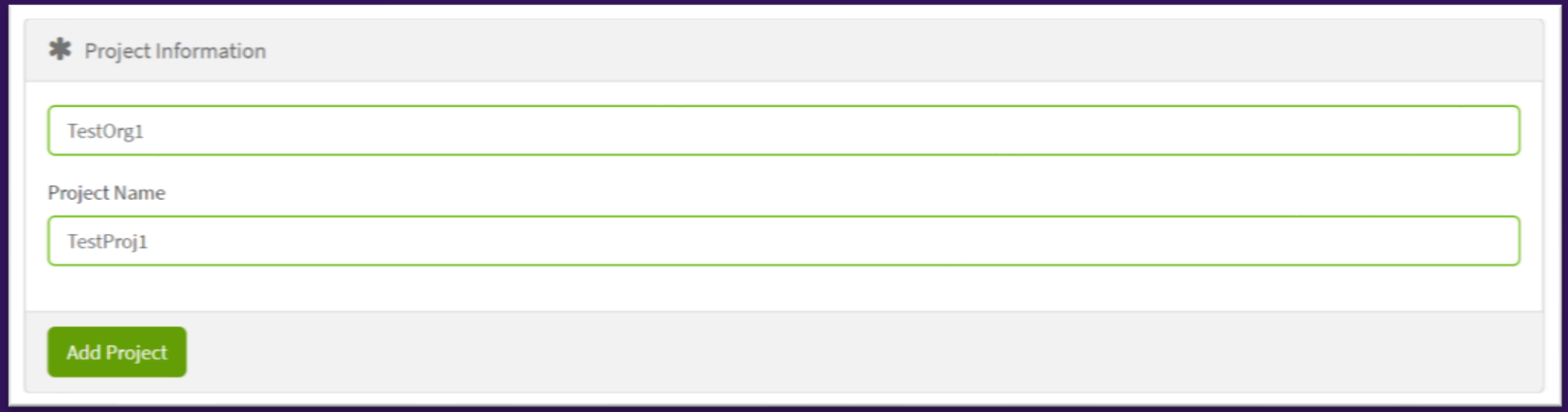
帳戶申請(1)


Exceptionless
to be without exception
Log in
Email
Email Address
Email Address is required.
Password
Password
Log in
Signup
Forgot password?
By signing up, you agree to our Privacy Policy and Terms of Service.

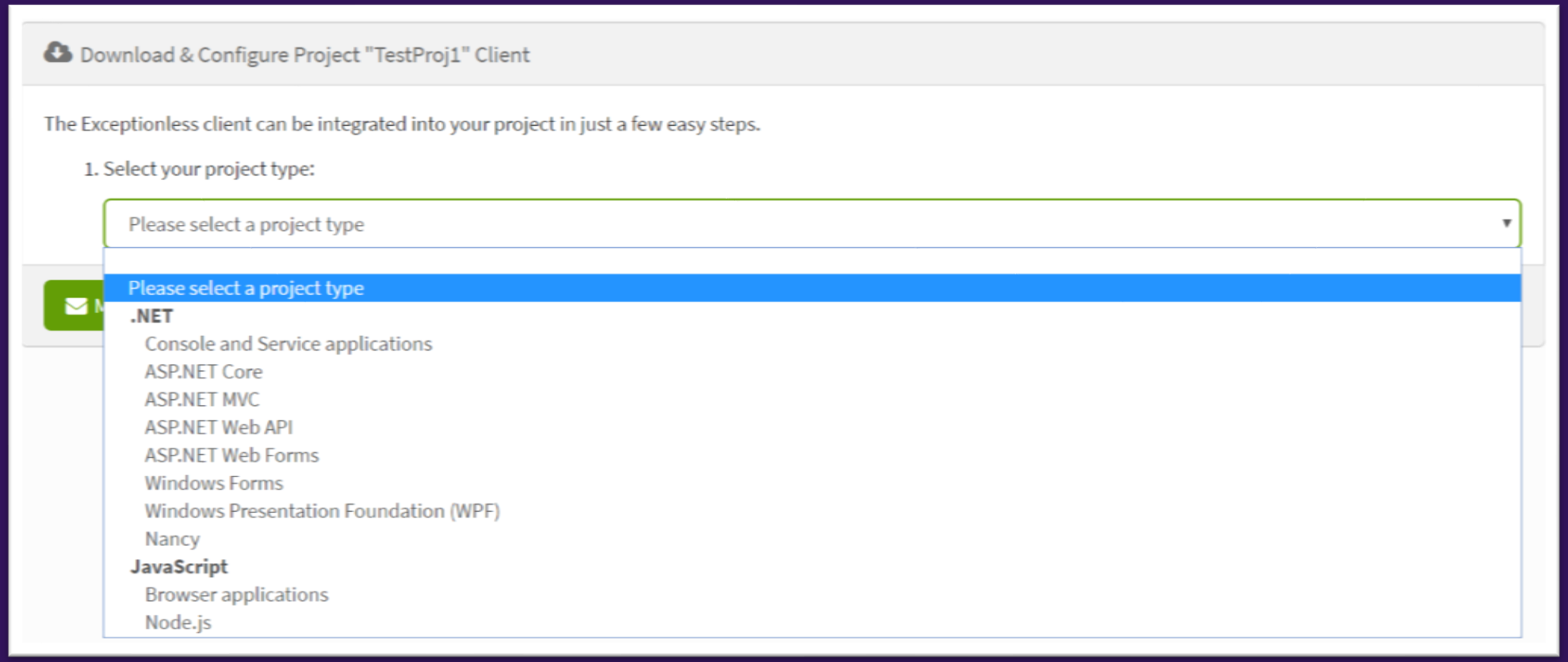
帳戶申請(2)


Exceptionless
to be without exception
Signup for a FREE account in seconds
Create an account
Name
Your first and last name
Full Name is required.
Email
Email address (no spam)
Password
Password
Create My Account
Already have an account? Log In
By signing up, you agree to our Privacy Policy and Terms of Service.

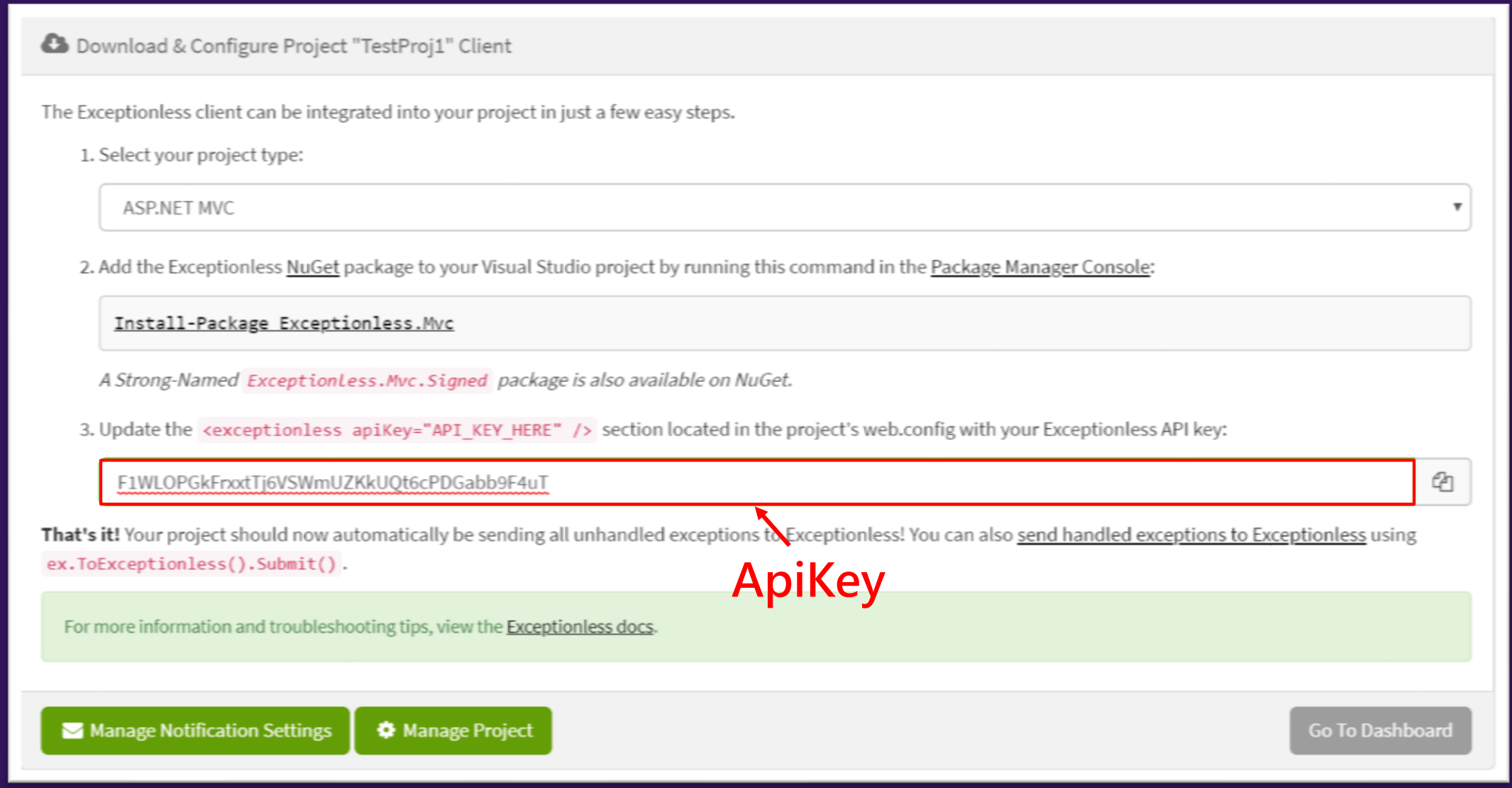
建立組織及專案(1)


Project Information
TestOrg1
Project Name
TestProj1
Add Project

建立組織及專案(2)


Download & Configure Project "TestProj1" Client
The Exceptionless client can be integrated into your project in just a few easy steps.
1. Select your project type:
Please select a project type
Please select a project type
.NET
Console and Service applications
ASP.NET Core
ASP.NET MVC
ASP.NET Web API
ASP.NET Web Forms
Windows Forms
Windows Presentation Foundation (WPF)
Nancy
JavaScript
Browser applications
Node.js

建立組織及專案(3)


Download & Configure Project "TestProj1" Client
The Exceptionless client can be integrated into your project in just a few easy steps.
1. Select your project type:
ASP.NET MVC
2. Add the Exceptionless NuGet package to your Visual Studio project by running this command in the Package Manager Console:
Install-Package Exceptionless.Mvc
A Strong-Named Exceptionless.Mvc.Signed package is also available on NuGet.
3. Update the <exceptionless apiKey="API_KEY_HERE" /> section located in the project's web.config with your Exceptionless API key:
F1WLOPGkFrxxtTj6VSWmUZKkUQt6cPDGabb9F4uT
That's it! Your project should now automatically be sending all unhandled exceptions to Exceptionless! You can also send handled exceptions to Exceptionless using ex.ToExceptionless().Submit().
ApiKey
For more information and troubleshooting tips, view the Exceptionless docs.
Manage Notification Settings
Manage Project
Go To Dashboard

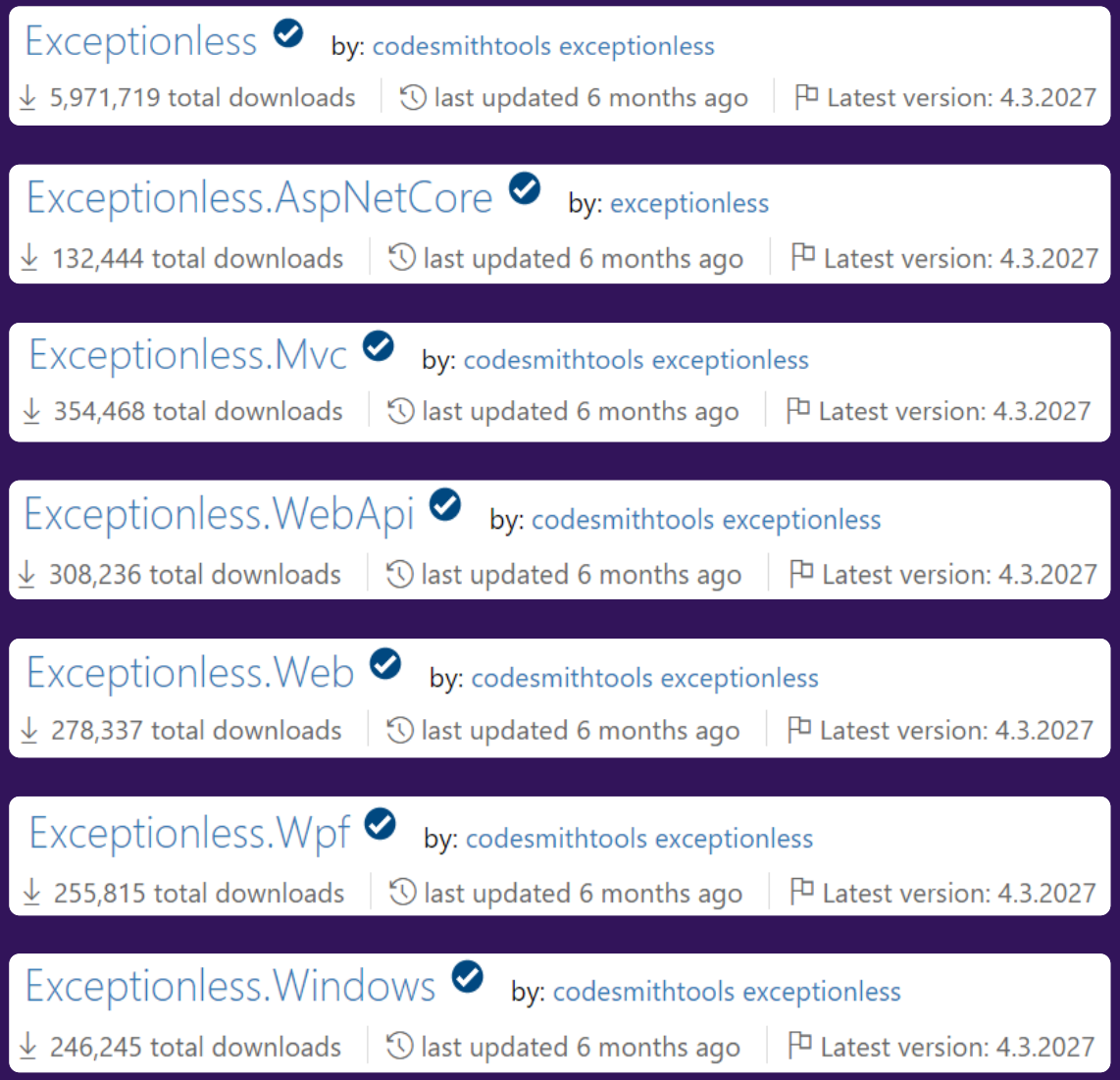


Exceptionless Client 篇

- 以 ASP.Net MVC5 為例

主要套件及支援框架(C#)

- Exceptionless
- ASP.Net Core
- ASP.Net MVC
- ASP.Net Web API
- Web Form
- WPF
- WinForm



Log支援套件

- NLog
- Log4Net
- SeriLog
- 自訂Log事件

Exceptionless.NLog by: codesmithtools exceptionless
75,548 total downloads | last updated 6 months ago | Latest version: 4.3.2027

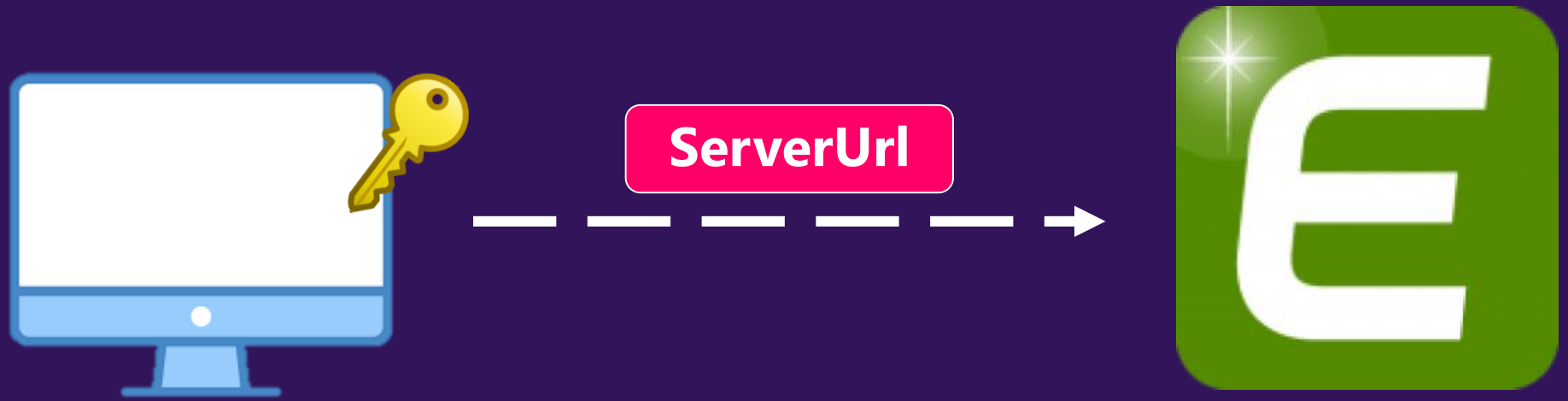
Exceptionless.Log4net by: exceptionless
62,257 total downloads | last updated 6 months ago | Latest version: 4.3.2027

Serilog.Sinks.ExceptionLess by: enkafan exceptionless niemyjski
78,937 total downloads | last updated 2017/11/6 | Latest version: 3.1.0



開始前請注意!!

- 請先正確設定 Api Key 及 ServerUrl 後再開始探索
- 請先正確設定 Api Key 及 ServerUrl 後再開始探索
- 請先正確設定 Api Key 及 ServerUrl 後再開始探索



組態設定(1) – Web.config

- 輸入Exceptionless Server上的Api Key
- 主要有3個區段要進行配置

```
<configuration>
  <configSections>
    <section name="exceptionless" type="Exceptionless.ExceptionlessSection, Exceptionless" />
  </configSections>

  <exceptionless apiKey="your server api key" serverUrl="your server url" />
  ...
  <system.webServer>
    <modules>
      <remove name="ExceptionlessModule" />
      <add name="ExceptionlessModule" type="Exceptionless.Mvc.ExceptionlessModule, Exceptionless.Mvc" />
    </modules>
    ...
  </system.webServer>
</configuration>
```

組態設定(2) – 程式碼呼叫

- 使用ExceptionlessClient.Default.Configuration設定
- 請於程式起始階段進行設定

```
//Global.asax.cs
public class MvcApplication : System.Web.HttpApplication
{
    protected void Application_Start()
    {
        AreaRegistration.RegisterAllAreas();
        FilterConfig.RegisterGlobalFilters(GlobalFilters.Filters);
        RouteConfig.RegisterRoutes(RouteTable.Routes);
        BundleConfig.RegisterBundles(BundleTable.Bundles);

        ExceptionlessClient.Default.Configuration.ApiKey = "your server api key";
        ExceptionlessClient.Default.Configuration.ServerUrl = "your server url";
    }
}
```

組態設定(3) – Assembly Attribute

- 需於startup被呼叫才會生效

```
using Exceptionless.Configuration;
[assembly: Exceptionless("your server api key", ServerUrl = "your server url")]
```

- 也可於起始階段另外宣告

```
ExceptionlessClient.Default.Configuration.
    ReadFromAttributes(typeof(MyClass).Assembly);
```

發送Exception(1)

- 預設會自動發送沒被處理的Exception
- 手動發送也是可以的

```
try
{
    throw new InvalidOperationException("please check your operation");
}
catch (Exception ex)
{
    ex.ToExceptionless().Submit();
}
```

發送Exception(2) - 客製額外資訊

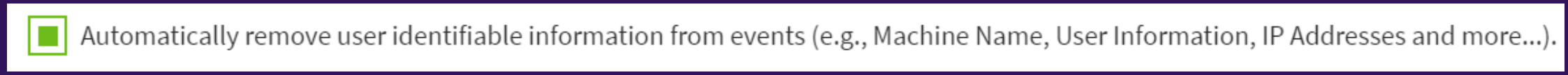
```
try
{
    throw new ApplicationException("Unable to create order from quote.");
}
catch (Exception ex)
{
    ex.ToExceptionless()
        .SetReferenceId(Guid.NewGuid().ToString("N"))
        .AddObject(order, "Order", excludedPropertyNames: new[] { "CreditCardNumber" }, maxDepth: 2)
        .SetProperty("Quote", 123)
        .AddTags("Order")
        .MarkAsCritical()
        .SetGeo(43.595089, -88.444602)
        .Submit();
}
```

GDPR(General Data Protection Regulation)

- 自動排除預設敏感資訊

```
ExceptionlessClient.Default.Configuration.IncludePrivateInformation = false;
```

- 可從Server>Project>Settings>Data Exclusions調整



- 依需求逐項設定

```
ExceptionlessClient.Default.Configuration.IncludeUserName = false;
ExceptionlessClient.Default.Configuration.IncludeMachineName = false;
ExceptionlessClient.Default.Configuration.IncludeIpAddress = false;
ExceptionlessClient.Default.Configuration.IncludeCookies = false;
ExceptionlessClient.Default.Configuration.IncludePostData = false;
ExceptionlessClient.Default.Configuration.IncludeQueryString = false;
```

發送Log(1) - Manually

- Log套件不支援時
- 自行開發Log套件時
- 可自訂額外資訊

```
ExceptionlessClient.Default.SubmitLog("message");
ExceptionlessClient.Default.SubmitLog("source", "message", LogLevel.Info).Submit();
ExceptionlessClient.Default.CreateLog("source", "message", LogLevel.Info)
    .AddTags(".net conf 2019").Submit();
```

發送Log (2) – NLog



- 安裝Exceptionless.NLog
  - ✓NLog
  - ✓NLog.Config
- 設定ApiKey及ServerUrl
- 設定對應的<target>及<rule>
- 預設包含PII(Personal Identifiable Information)資訊

發送Log(3) – NLog.config

```
<nlog ...>
  <extensions>
    <add assembly="Exceptionless.NLog"/>
  </extensions>
  <targets async="true">
    <target xsi:type="Exceptionless" name="exceptionless"
            apiKey="your server api key" serverUrl="your server url">
      <field name="host" layout="${machinename}" />
      <field name="identity" layout="${identity}" />
      <field name="windows-identity" layout="${windows-identity:userName=True:domain=False}" />
      <field name="process" layout="${processname}" />
    </target>
  </targets>
  <rules>
    <logger name="*" minlevel="Trace" writeTo="exceptionless" />
  </rules>
</nlog>
```

發送Log(4) – Use NLog

```
public ActionResult NLogTest()
{
    var logger = LogManager.GetCurrentClassLogger();

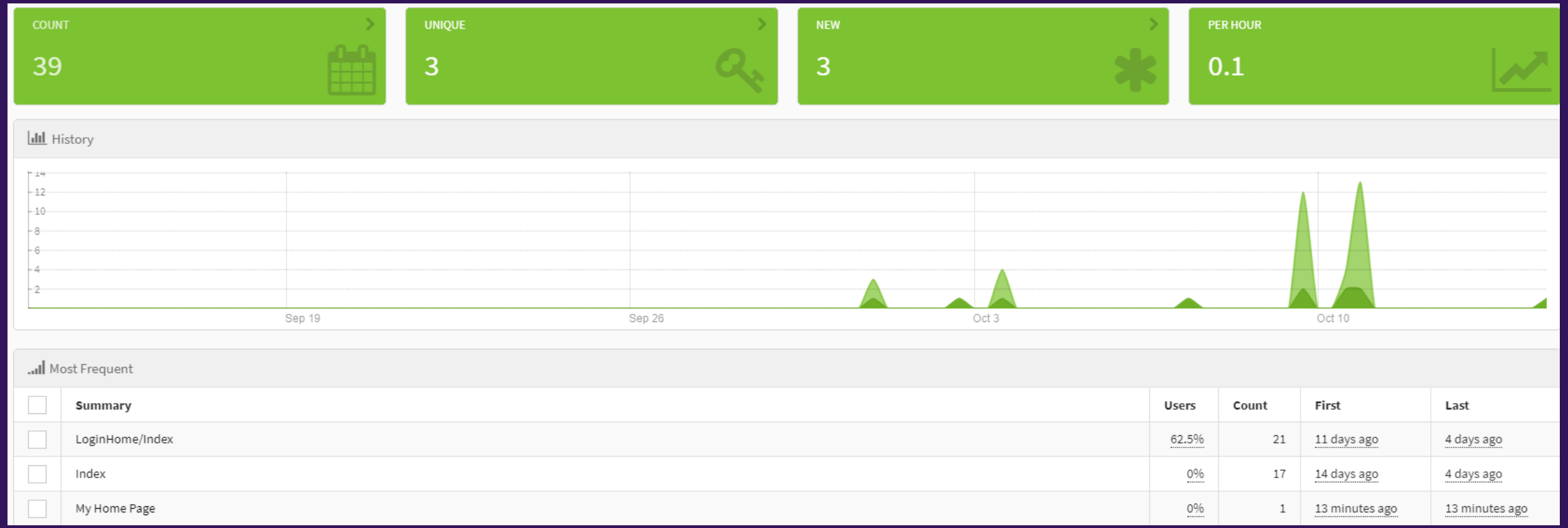
    logger.Trace("NLog - Trace in Mvc5");
    logger.Debug("NLog - Debug in Mvc5");
    logger.Info("NLog - Info in Mvc5");
    logger.Warn("NLog - Warn in Mvc5");
    logger.Error("NLog - Error in Mvc5");
    logger.Fatal("NLog - Fatal in Mvc5");

    return View();
}
```

Feature Usage(1)

- 統計使用功能次數資訊

```
ExceptionlessClient.Default.CreateFeatureUsage("My Home Page")
    .AddTags("mvc5").Submit();
```



Feature Usage(2) – 嫌麻煩?

- 召喚 ActionFilter → 經濟又實惠

```
public class FilterConfig
{
    0 references
    public static void RegisterGlobalFilters(GlobalFilterCollection filters)
    {
        filters.Add(new HandleErrorAttribute());
        filters.Add(new FeatureUsageAttribute());
    }
}

}
```

User Sessions(1)

- 於發送事件中加上使用者資訊
- 會自動嘗試獲取使用者資訊
- 預設每30秒偵測一次HeartBeat
- 搭配Feature Usage使用 → 使用者行為分析

```
ExceptionlessClient.Default.Configuration.SetUserIdentity("UserId", "UserName");
ExceptionlessClient.Default.Configuration.UseSessions();
```

```
public static void UseSessions(this ExceptionlessConfiguration config,
    bool sendHeartbeats = true, TimeSpan? heartbeatInterval = null,
    bool useSessionIdManagement = false)
```

User Sessions(2)

- 修改Heartbeat間隔

```
ExceptionlessClient.Default.Configuration.
        UseSessions(true, TimeSpan.FromMinutes(5));
```

- Disable Heartbeat

```
ExceptionlessClient.Default.Configuration.UseSessions(false);
```

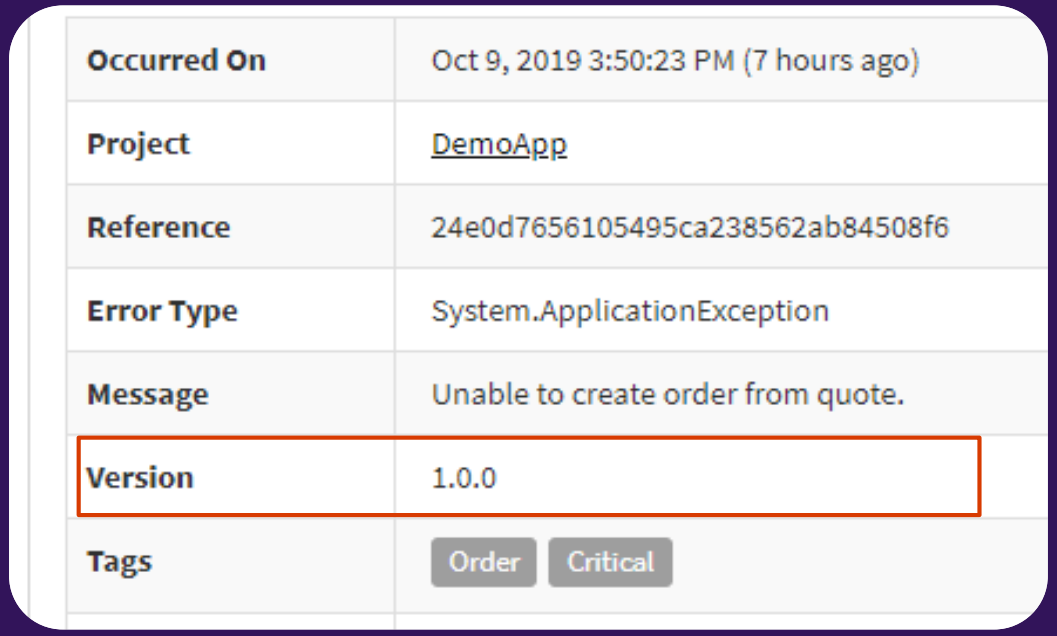


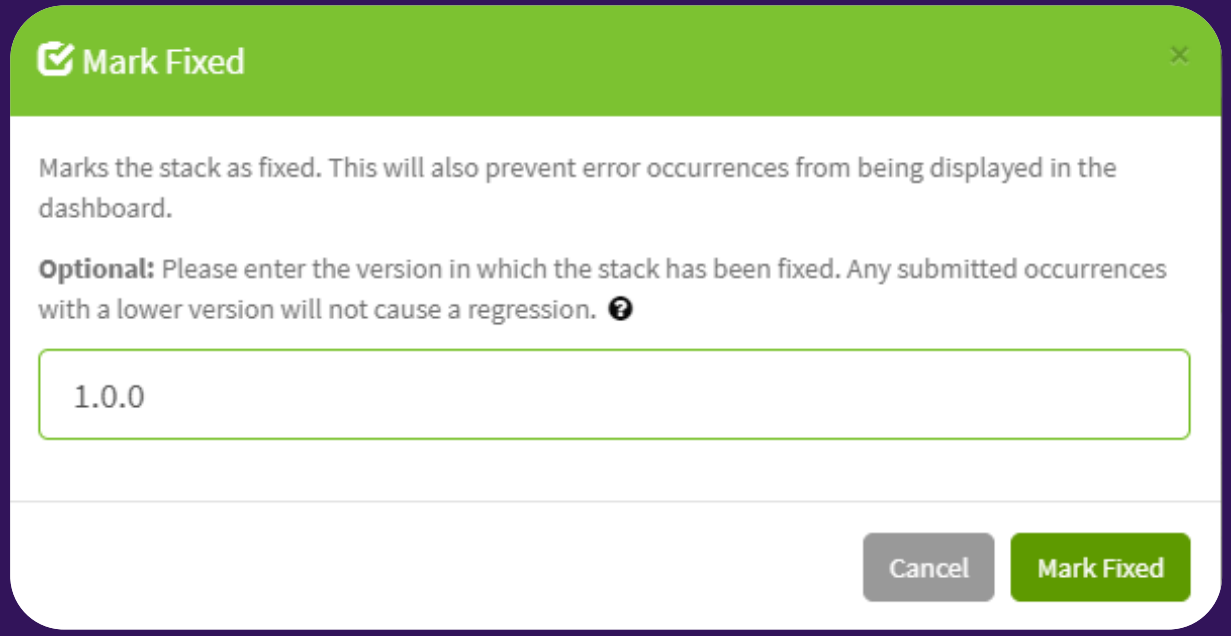
Exceptionless 番外篇

Versioning

- 可註記當前應用程式**版號**
- 可搭配**Mark Fixed**區別錯誤事件

```
ExceptionlessClient.Default.Configuration.SetVersion("1.0.0");
```





組態管理(1)

- 開發及測試過程中不傳送錯誤
- 官方提供enabled開關功能(不推薦)
- 依環境轉換對應的組態檔(Dev、Test、Production)
- 使用VS內建Configuration Transform進行轉換
- NLog.config需額外安裝**SlowCheetah**擴充套件

code
```
ExceptionlessClient.Default.Configuration.Enabled = false;
```

config
```
<exceptionless apiKey="YOUR_API_KEY" enabled="false" />
```

組態管理(2) – Web.Release.config

- 使用Transform的**Replace**模式進行轉換

Web.config

```
<exceptionless apiKey="your server api key" />
```

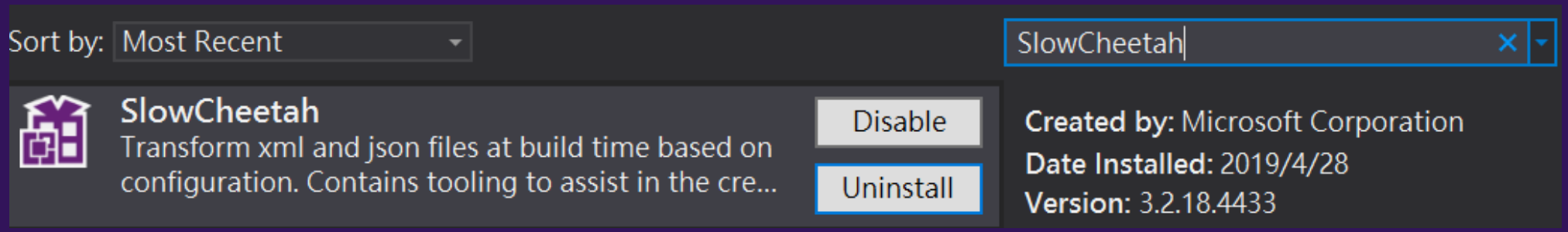
Web.Release.config

```
<configuration xmlns:xdt="http://schemas.microsoft.com/XML-Document-Transform">
  ...
  <exceptionless apiKey="your server api key"
      serverUrl="http://localhost:5000" xdt:Transform="Replace" />
  ...
</configuration>
```

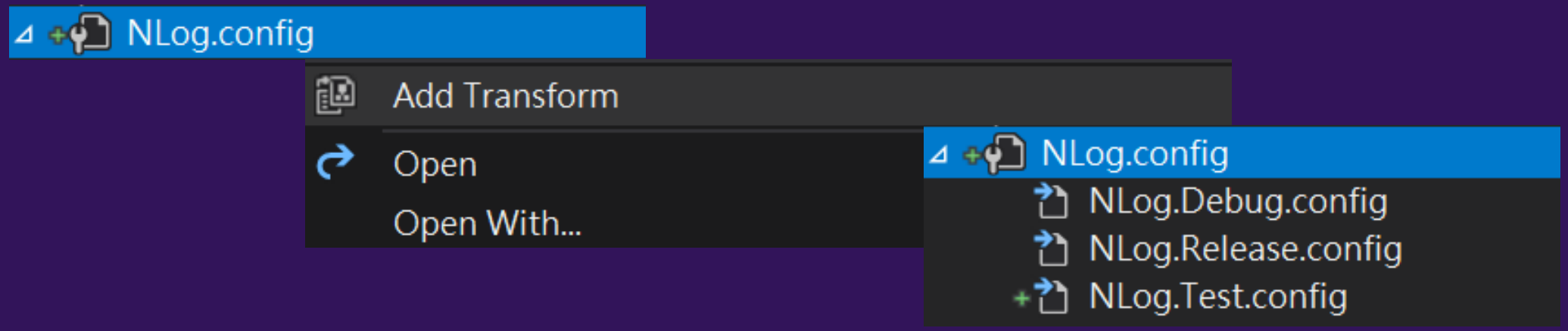
完成後可在右鍵選單內點選Preview Transform進行組態比較

組態管理(3) – 安裝SlowCheetah

- 於Visual Studio Marketplace安裝**SlowCheetah**



- 於**NLog.config**按右鍵並點選**Add Transform**



組態管理(4) – NLog.Release.config

- 使用Transform的SetAttributes模式進行轉換

NLog.config

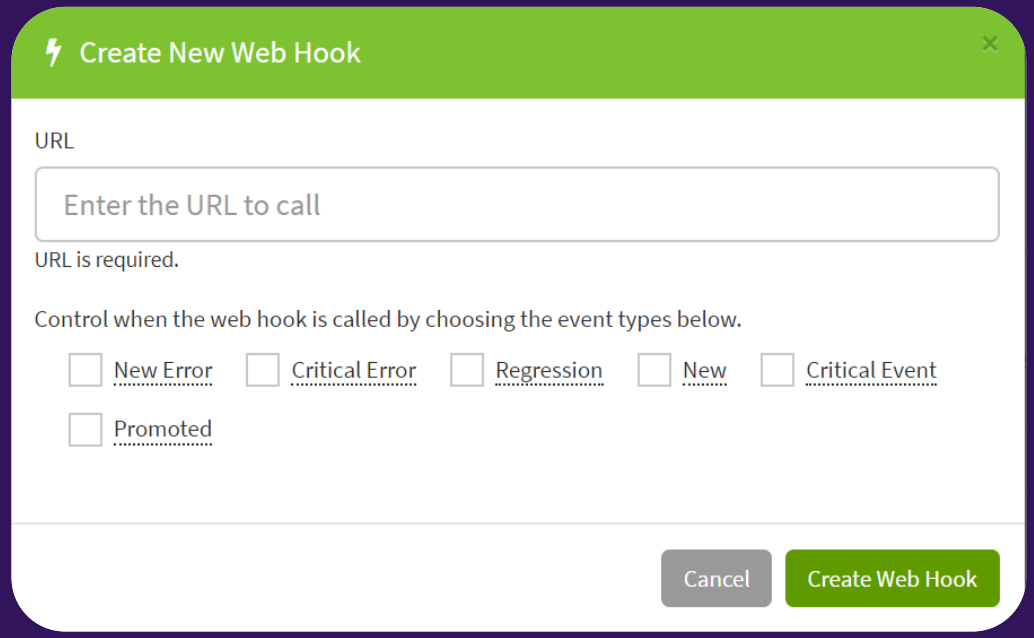
```
<target xsi:type="Exceptionless" name="exceptionless">
  <field ... />
</target>
```

NLog.Release.config

```
<nlog ...>
  <targets async="true">
    <target xsi:type="Exceptionless" name="exceptionless"
        apiKey="your server api key" serverUrl="http://localhost:5000"
        xdt:Transform="SetAttributes" xdt:Locator="Match(name)"/>
  </targets>
</nlog>
```

Integrations(1)

- 將事件訊息轉送至外部系統
- 提供外部系統介接整合彈性
- 可設定事件轉送條件





Integrations(2)

Web hooks

The following web hooks will be called for this project. We also support Zapier which allows you to easily integrate Exceptionless with many different services (750+) like JIRA, HipChat, Twilio, Basecamp and more.

Url	Event Types	Actions
http://localhost:26200/External/Integration	StackPromoted StackRegression	×

```
[HttpPost]
public void Integration()
{
    var request = Request.InputStream;
    request.Seek(0, System.IO.SeekOrigin.Begin);
    var json = new StreamReader(request).ReadToEnd();
    var data = JsonConvert.DeserializeObject<ExceptionlessWebHookModel>(json);
}
```

Integrations(3)

[JSON]
id: "5d9e0c34b728d00001f2992d"
url: "http://ex-ui.localtest.me:5100/stack/5d9e0c34b728d00001f2992d"
title: "方法或作業尚未實作。"
type: "System.NotImplementedException"
target_method: "ExceptionlessMvc5Client.Services.MyService.Common()"
project_id: "5d6616ccc752230010147ec"
project_name: "DemoApp"
organization_id: "5d6616ccc752230010147eb"
organization_name: "MyOrg"
total_occurrences: 11
first_occurrence: "2019/10/10 上午 12:34:58"
last_occurrence: "2019/10/10 下午 06:46:33"
date_fixed: "2019/10/10 下午 06:46:22"
is_regression: True
is_critical: False

Data Exclusions

- 可設定欲排除的資料key值
- 支援萬用字元(*, Wild Cards)搜尋

Error Stacking

- 用Hash(ExceptionType+Method)當堆疊目標依據
- 設定User Namespaces可以限制哪些命名空間底下的錯誤才要視為堆疊目標
- 設定Common Method則是限制哪些方法內的錯誤不要視為堆疊目標

Client Configuration(1)

- 可從Server端改變Client端的事件發送行為
- 程式初始後隔5秒會更新第一次組態設定
- 最後一個事件發送後每2分鐘進行更新
- Client端透過EventPlugin處理事件的行為
- 提供使用者加入自訂的Plugin

```
//Server會將Client Configuration同步到Client的SettingsDictionary
ExceptionlessClient.Default.Configuration.Settings;
```

```
SettingsDictionary ExceptionlessConfiguration.Settings { get; }
Contains a dictionary of custom settings that can be used to control the client and will be automatically updated from the server.
```

Client Configuration(2) - Event Exclusions

- 可排除error, usage, log, 404, session等事件
- Event Exclusions就是一種EventPlugin
- 使用情境可參考官方GitHub
- 例如：排除**NullReferenceException**的錯誤事件

Key	Value	Actions
@@error:System.NullReferenceException	false	Edit ✖

自訂EventPlugin(1)

- 提供兩種實作方式
  - ✓使用委派方法 : AddPlugin(key, priority, eventContext =>{ ... })
  - ✓實作IEventPlugin後再手動加入 : AddPlugin<IEventPlugin>()
- 搭配Client Configuration彈性調整組態
- 加入及移除EventPlugin

```
ExceptionlessClient.Default.Configuration.AddPlugin();
ExceptionlessClient.Default.Configuration.RemovePlugin();
```

自訂EventPlugin(2) – 使用委派

```
ExceptionlessClient.Default.Configuration.AddPlugin("cancel log submission", 100, context =>
    {
        var enableLogSubmission = context.Client.Configuration.Settings
            .GetBoolean("enableLogSubmission", true);

        //當enableLogSubmission為false時會取消Log事件發送
        if (context.Event.Type == Event.KnownTypes.Log && !enableLogSubmission)
        {
            context.Cancel = true;
        }
    });
```

自訂EventPlugin(3) – 實作IEventPlugin

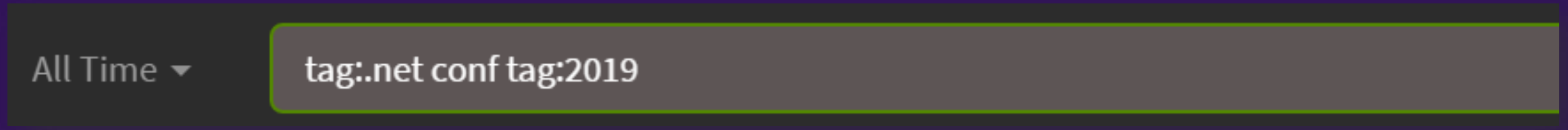
```
[Priority(100)]
public class LogSubmissionPlugin : IEventPlugin
{
    public void Run(EventPluginContext context)
    {
        var enableLogSubmission = context.Client.Configuration.Settings.
                GetBoolean("enableLogSubmission", true);

        //當enableLogSubmission為false時會取消Log事件發送
        if (context.Event.Type == Event.KnownTypes.Log && !enableLogSubmission)
        {
            context.Cancel = true;
        }
    }
}
```

```
ExceptionlessClient.Default.Configuration.AddPlugin<LogSubmissionPlugin>();
```

Filter & Searching

- 支援萬用字元(* ,Wild Cards)搜尋
- 多條件搜尋
  - ✓預設為**AND**，請用空白符號分隔(**tag:dotnetconf tag:2019**)
  - ✓**OR**搜尋請用OR字串隔開(**tag:dev OR tag:ops**)
- 細部資訊請參考作者GitHub文件



docker-compose設定

- 主要分為api、jobs、ui 三塊
- 設定細節請參考作者GitHub
- 開頭必須加入**EX_**以供識別
- 調整資料最大保存天數
  - ✓**EX_MaximumRetentionDays: 30**
- 調整單一事件最大傳送(bytes)
  - ✓**EX_MaximumEventPostSize : 4096**

官方文件參考

- 英文：https://github.com/Exceptionless/Exceptionless/wiki
- 中文(進行中)：https://github.com/wtx2043pp986/Exceptionless/wiki
- 範例程式碼：https://github.com/wtx2043pp986/ExceptionlessSample/commits/master



https://skilltree.my

無界開發者社群

現在開發者已經不能單純的只會一種技術了，這社群以 .NET 為主，其他技術為輔，寫程式是要解決問題的，不是用程式語言框住自己，看看駐站版主群陣容多龐大!



天瓏資訊圖書





Free Personal Subscription 100% DISCOUNT CODE
AppCode, CLion, DataGrip, GoLand, IntelliJ IDEA Ultimate, PhpStorm, PyCharm, ReSharper, ReSharper C++, ReSharper Ultimate, ReSharper Ultimate + Rider Pack, Rider, RubyMine, or WebStorm