

# .NET Development Challenge



## Introduction

The Empathica .NET Development Challenge is intended to gauge how quickly development candidates can learn relevant technical skills. You will **not** be evaluated purely on successful completion of all challenges, but instead our goal is to discover how effectively we can work as a team to make great software.

These challenges are progressively more difficult and touch on many of the core technical tools and concepts required for building a modern web application. For some candidates without existing knowledge of web applications and databases these challenges may be very difficult.

Upon completion of the challenge you will be asked to bring in your work, demonstrate the challenges you were able to successfully complete, explain the technologies and how they work, and work with a team of Empathica engineers improve your software.

## Challenge 1: Basic tools

Install the suite of .NET web development tools from (you may use Visual Studio if you already have it)

<http://www.microsoft.com/express/Web/>

All application development should be completed using MVC 2 for pages and the Entity Framework for database access.

## Challenge 2: Hello world wide web

Create a 'hello world' web page that runs on a local server using MVC 2.

Helpful tutorials are available here: <http://www.w3schools.com/>

## Challenge 3: Chat to the world wide web

Create a 'hello world' web form that has a text box, and a 'Done' button using MVC 2. When 'Done' is clicked the next page says 'Hello world' and below that includes the text that was previously typed into the text box.

Create a few unit tests to verify your chat application. You may use any unit test framework you prefer (typically either Visual Studio tests, or NUnit tests).

# .NET Development Challenge



## **Challenge 4: Twitter to the world**

Create a 'hello world' twitter application that has a text box and a 'Done' button using MVC 2 and any Javascript library you like. When 'Done' is clicked the text typed into the text box is posted to a Twitter account.

## **Challenge 5: Twitter to the world and a database**

Create a 'hello world' twitter application that has a text box and a 'Done' button and logs all input to a database using MVC 2 and the Entity Framework. Just like challenge 4, but also logs info to a database including the text of the tweet, the twitter id of the twitter account, and the date and time when the tweet was made.

## **Challenge 6: Configurable Twitter**

Extend the twitter application from challenge 5 by making all display labels (like the prompt above the tweet text box) configurable from a database. Cache the strings in memory so that you do not have to access the database every time you display the page.

## **Challenge 7: Twitter with short URL**

Extend your solution from challenge 6 by creating a page that displays a past tweet message given some identifier. The goal is to be able to include this link in the tweet, so the URL must be as short as possible. Ideally the URL to the past tweet page will be something like this:

<http://localhost/123>

where 123 would be some identifier for the past tweet.

## **Challenge 8: Twitter with an error page**

Web mashups can return weird errors. Add a friendly error page to your tweet application with text like 'Oops a problem occurred'. This error page should be displayed automatically for every page of your application in the event of an exception

## **Challenge 9: Secure your Twitter application with https**

Add https capability to the solution to challenge 8.