

برخی از تنظیمات پروژه نباید به مخازن سورس کنترل ارسال شوند؛ حال یا نیازی به این کار نیست یا مقادیر تنظیمات محرمانه هستند. چند بار پیش آمده‌است که پروژه را از سورس کنترل دریافت و مجبور شده باشید رشته‌های اتصال و دیگر تنظیمات را مجدداً ویرایش کنید، چرا که توسعه دهندگان دیگری مثلاً فایل‌های Web/App.config خود را به اشتباه push کرده اند؟ حتی اگر تنظیمات پروژه محرمانه هم نباشند (مثلاً پسورد دیتابیس‌ها یا ایمیل‌ها) این موارد می‌توانند دردسر ساز شوند. بدتر از اینها هنگامی است که تنظیمات محرمانه را به مخازنی عمومی (مثلاً GitHub) ارسال می‌کنید!

یک فایل web.config معمولی را در نظر بگیرید (اطلاعات غیر ضروری حذف شده اند).

```
<?xml version="1.0" encoding="utf-8"?>
<!--
  A bunch of ASP.NET MVC web config stuff goes here . . .
-->
<configuration>
  <connectionStrings>
    <add name="DefaultConnection" value="YourConnectionStringAndPassword"/>
  </connectionStrings>

  <appSettings file="PrivateSettings.config">
    <add key="owin:AppStartup"
value="AspNetIdentity2ExtendingApplicationUser.Startup,AspNetIdentity2ExtendingApplicationUser" />
    <add key="webpages:Version" value="3.0.0.0" />
    <add key="webpages:Enabled" value="false" />
    <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
    <add key="EMAIL_PASSWORD" value="YourEmailPassword"/>
  </appSettings>
</configuration>
```

در تنظیمات بالا یک رشته اتصال وجود دارد که ترجیحاً نمی‌خواهیم به سورس کنترل ارسال کنیم، و یا اینکه این رشته اتصال بین توسعه دهندگان مختلف متفاوت است. همچنین کلمه عبور یک ایمیل هم وجود دارد که نمی‌خواهیم به مخازن سورس کنترل ارسال شود، و مجدداً ممکن است مقدارش بین توسعه دهندگان متفاوت باشد. از طرفی بسیاری از تنظیمات این فایل متعلق به کل اپلیکیشن است، بنابراین صرفنظر کردن از کل فایل web.config در سورس کنترل گزینه جالبی نیست.

خوشبختانه کلاس ConfigurationManager راه حل هایی پیش پای ما می‌گذارد.

استفاده از خاصیت configSource برای انتقال قسمت هایی از تنظیمات به فایل مجزا

با استفاده از خاصیت configSource می‌توانیم قسمتی از تنظیمات (configuration section) را به فایل مجزا منتقل کنیم. بعنوان مثال، رشته‌های اتصال از مواردی هستند که می‌توانند بدین صورت تفکیک شوند.

بدین منظور می‌توانیم فایل تنظیمات جدیدی (مثلاً با نام connectionStrings.config) ایجاد کنیم و سپس با استفاده از خاصیت نام برده در فایل web.config به آن ارجاع دهیم. برای این کار فایل تنظیمات جدیدی ایجاد کنید و مقادیر زیر را به آن اضافه کنید (xml header یا هیچ چیز دیگری نباید در این فایل وجود داشته باشد، تنها مقادیر تنظیمات).

```
<connectionStrings>
  <add name="DefaultConnection" value="YourConnectionStringAndPassword"/>
</connectionStrings>
```

حال باید فایل web.config را ویرایش کنیم. رشته‌های اتصال را حذف کنید و با استفاده از خاصیت configSource تنها به فایل تنظیمات اشاره کنید.

```
<connectionStrings configSource="ConnectionStrings.config">
</connectionStrings>
```

دسترسی به رشته‌های اتصال مانند گذشته انجام می‌شود. به بیان دیگر تمام تنظیمات موجود (حال مستقیم یا ارجاع شده) همگی بصورت یکپارچه دریافت شده و به کد کلاینت تحویل می‌شوند.

```
var conn = ConfigurationManager.ConnectionStrings["DefaultConnection"];
string connString = conn.ConnectionString;
// etc.
```

در قطعه کد بالا، دسترسی به رشته‌های اتصال بر اساس نام، آبجکتی از نوع ConnectionStringSettings را بر می‌گرداند. خاصیت configSource برای هر قسمت از تنظیمات پیکربندی می‌تواند استفاده شود.

استفاده از خاصیت file برای انتقال بخشی از تنظیمات به فایلی مجزا

ممکن است فایل تنظیمات شما (مثلا web.config) شامل مقادیری در قسمت <appSettings> باشد که برای کل پروژه تعریف شده اند (global) اما برخی از آنها محرمانه هستند و باید از سورس کنترل دور نگاه داشته شوند. در این سناریوها خاصیتی بنام file وجود دارد که مختص قسمت appSettings است و به ما اجازه می‌دهد مقادیر مورد نظر را به فایلی مجزا انتقال دهیم. هنگام دسترسی به مقادیر این قسمت تمام تنظیمات بصورت یکجا خوانده می‌شوند.

در مثال جاری یک کلمه عبور ایمیل داریم که می‌خواهیم محرمانه بماند. بدین منظور می‌توانیم فایل پیکربندی جدیدی مثلا با نام PrivateSettings.config ایجاد کنیم. این فایل هم نباید xml header یا اطلاعات دیگری داشته باشد، تنها مقادیر appSettings را در آن نگاشت کنید.

```
<appSettings>
  <add key="MAIL_PASSWORD" value="xspbqmurkjadeck"/>
</appSettings>
```

حال تنظیمات کلمه عبور را از فایل web.config حذف کنید و با استفاده از خاصیت file، به فایل جدید اشاره کنید.

```
<appSettings file="PrivateSettings.config">
  <add key="owin:AppStartup"
value="AspNetIdentity2ExtendingApplicationUser.Startup,AspNetIdentity2ExtendingApplicationUser" />
  <add key="webpages:Version" value="3.0.0.0" />
  <add key="webpages:Enabled" value="false" />
  <add key="ClientValidationEnabled" value="true" />
  <add key="UnobtrusiveJavaScriptEnabled" value="true" />
</appSettings>
```

دسترسی به تنظیمات appSettings مانند گذشته انجام می‌شود. همانطور که گفته شد ConfigurationManager بصورت خودکار اینگونه ارجاعات را تشخیص داده و تمام اطلاعات را بصورت یکجا در اختیار client code قرار می‌دهد.

```
var pwd = ConfigurationManager.AppSettings["MAIL_PASSWORD"];
```

فایل‌های ویژه را به gitignore اضافه کنید

حال می‌توانیم فایل web.config را به سورس کنترل اضافه کنیم، فایل‌های ConnectionStrings.config و PrivateSettings.config را به فایل gitignore اضافه کنیم و پروژه را commit کنیم. در این صورت فایل‌های تنظیمات خصوصی به مخازن سورس کنترل ارسال نخواهند شد.

مستند سازی را فراموش نکنید!

مسئله اگر چنین رویکردی را در پیش بگیرید باید دیگران را از آن مطلع کنید (مثلا با افزودن توضیحاتی به فایل README.txt). بهتر است در فایل web.config خود هر جا که لازم است توضیحات XML خود را درج کنید و به توسعه دهندگان توضیح دهید که چه فایل هایی را روی نسخه های محلی خود باید ایجاد کنند و هر کدام از این فایل ها چه محتوایی باید داشته باشند.

نظرات خوانندگان

نویسنده: رضایی
تاریخ: ۱۳۹۳/۰۲/۰۸ ۰:۱۲

با سلام؛ ممنون بابت مطلب مفیدتون.
میشه در خصوص gitignore توضیحاتی بفرمایید؟

نویسنده: وحید نصیری
تاریخ: ۱۳۹۳/۰۲/۰۸ ۰:۱۷

توضیحات بیشتر در سری Git » [آموزش سیستم مدیریت کد Git : استفاده به صورت محلی \(بخش دوم\)](#) «