



TS Web Applications Development with Microsoft .NET Framework 4 - Exam 70-515

About this Exam

This exam is designed to test the candidate's knowledge and skills for developing Web applications using ASP.NET and the .NET Framework 4. Questions that contain code will be presented in either VB or C#. Candidates can select one of these languages when they start the exam.



How long is this course?

2KO International presents this course as a [five day full time course](#).

We occasionally run this course as a part time course over 5 Saturdays (based on demand)

Where does the course take place?

At our training centre in Greenpoint, Cape Town, South Africa.

Or we can run it in-house at your business, anywhere in the world (requires individualised quotation).



We aim to offer training at the best prices – if you get a cheaper quote than us, send it to us (official quotes only) and we will beat it. Simple!

Audience Profile

- Candidates for this exam are professional Web developers who use Microsoft Visual Studio.
- Candidates should have a minimum of two to three years of experience developing Web-based applications by using Visual Studio and Microsoft ASP.NET.
- Candidates should be experienced users of Visual Studio 2008 and later releases and should have a fundamental knowledge of the .NET Framework 4 programming languages (C# or Microsoft Visual Basic). In addition, candidates should understand how to use the new features of Visual Studio 2010 and the .NET Framework 4.
- Candidates should also have a minimum of one year of experience with the following:
 - Accessing data by using Microsoft ADO.NET and LINQ



- Creating and consuming Web and Windows Communication Foundation (WCF) services
- State management
- ASP.NET configuration
- Debugging and deployment
- Application and page life-cycle management
- Security aspects such as authentication and authorization
- Client-side scripting languages
- Internet Information Server (IIS)
- ASP.NET MVC

Skills Being Measured

This exam measures your ability to accomplish the technical tasks listed below. The percentages indicate the relative weight of each major topic area on the exam.

Developing Web Forms Pages (19%)

- Configure Web Forms pages.
This objective may include but is not limited to: page directives such as ViewState, request validation, event validation, MasterPageFile; ClientIDMode; using web.config; setting the html doctype
This objective does not include: referencing a master page; adding a title to a Web form
- Implement master pages and themes.
This objective may include but is not limited to: creating and applying themes; adding multiple content placeholders; nested master pages; control skins; passing messages between master pages; switching between themes at runtime; loading themes at run time; applying a validation schema. This objective does not include: creating a master page; basic content pages
- Implement globalization.
This objective may include but is not limited to: resource files, browser files, CurrentCulture, currentUICulture, ASP:Localize
- Handle page life cycle events.
This objective may include but is not limited to: IsPostBack, IsValid, dynamically creating controls, control availability within the page life cycle, accessing control values on postback, overriding page events
- Implement caching.
This objective may include but is not limited to: data caching; page output caching; control output caching; cache dependencies; setting cache lifetimes; substitution control
This objective does not include: distributed caching (Velocity)
- Manage state.
This objective may include but is not limited to: server-side technologies, for example, session and application; client-side technologies, for example, cookies and ViewState; configuring session state (in proc, state server, Microsoft SQL



Server; cookieless); session state compression; persisting data by using ViewState; compressing ViewState; moving ViewState

Developing and Using Web Forms Controls (18%)

- Validate user input.

This objective may include but is not limited to: client side, server side, and via AJAX; custom validation controls; regex validation; validation groups; datatype check; jQuery validation

This objective does not include: RangeValidator and RequiredValidator

- Create page layout.

This objective may include but is not limited to: AssociatedControlID; Web parts; navigation controls; FileUpload controls

This objective does not include: label; placeholder, panel controls; CSS, HTML, referencing CSS files, inlining

- Implement user controls.

This objective may include but is not limited to: registering a control; adding a user control; referencing a user control; dynamically loading a user control; custom event; custom properties; setting toolbox visibility

- Implement server controls.

This objective may include but is not limited to: composite controls, INamingContainer, adding a server control to the toolbox, global assembly cache, creating a custom control event, globally registering from web.config; TypeConverters

This objective does not include: postback data handler, custom databound controls, templated control

- Manipulate user interface controls from code-behind.

This objective may include but is not limited to: HTML encoding to avoid cross-site scripting, navigating through and manipulating the control hierarchy; FindControl; controlRenderingCompatibilityVersion; URL encoding; RenderOuterTable

This objective does not include: Visibility, Text, Enabled properties

Implementing Client-Side Scripting and AJAX (16%)

- Add dynamic features to a page by using JavaScript.

This objective may include but is not limited to: referencing client ID; Script Manager; Script combining; Page.clientscript.registerclientscriptblock; Page.clientscript.registerclientscriptinclude; sys.require (scriptloader)

This objective does not include: interacting with the server; referencing JavaScript files; inlining JavaScript

- Alter a page dynamically by manipulating the DOM.

This objective may include but is not limited to: using jQuery, adding, modifying, or removing page elements, adding effects, jQuery selectors

This objective does not include: AJAX

- Handle JavaScript events.

This objective may include but is not limited to: DOM events, custom events, handling events by using jQuery



- Implement ASP.NET AJAX.

This objective may include but is not limited to: client-side templating, creating a script service, extenders (ASP.NET AJAX Control Toolkit), interacting with the server, Microsoft AJAX Client Library, custom extenders; multiple update panels; triggers; UpdatePanel.UpdateMode; Timer

This objective does not include: basic update panel and progress

- Implement AJAX by using jQuery.

This objective may include but is not limited to: \$.get, \$.post, \$.getJSON, \$.ajax, xml, html, JavaScript Object Notation (JSON), handling return types

This objective does not include: creating a service

Configuring and Extending a Web Application (15%)

- Configure authentication and authorization.

This objective may include but is not limited to: using membership, using login controls, roles, location element, protecting an area of a site or a page

This objective does not include: Windows Live; Microsoft Passport; Windows and Forms authentication

- Configure providers.

This objective may include but is not limited to: role, membership, personalization, aspnet_regsql.exe

This objective does not include: creating custom providers

- Create and configure HttpHandlers and HttpModules.

This objective may include but is not limited to: generic handlers, asynchronous handlers, setting MIME types and other content headers, wiring modules to application events

- Configure initialization and error handling.

This objective may include but is not limited to: handling Application_Start, Session_Start, and Application_BeginRequest in global.asax, capturing unhandled exceptions, custom error section of web.config, redirecting to an error page; try and catch; creating custom exceptions

- Reference and configure ASMX and WCF services.

This objective may include but is not limited to: adding service reference, adding Web reference, changing endpoints, wsdl.exe, svcutil.exe; updating service URL; shared WCF contracts assembly

This objective does not include: creating WCF and ASMX services

- Configure projects and solutions, and reference assemblies.

This objective may include but is not limited to: local assemblies, shared assemblies (global assembly cache), Web application projects, solutions, settings file, configuring a Web application by using web.config or multiple .config files; assemblyinfo

- Debug a Web application.

This objective may include but is not limited to: remote, local, JavaScript debugging, attaching to process, logging and tracing, using local IIS, aspnet_regiis.exe

- Deploy a Web application.



This objective may include but is not limited to: pre-compilation, publishing methods (e.g.,

MSDeploy, xcopy, and FTP), deploying an MVC application

This objective does not include: application pools, IIS configuration

Displaying and Manipulating Data (19%)

- Implement data-bound controls.

This objective may include but is not limited to: advanced customization of DataList, Repeater, ListView, FormsView, DetailsView, TreeView, DataPager, Chart, GridView

This objective does not include: working in Design mode

- Implement DataSource controls.

This objective may include but is not limited to: ObjectDataSource, LinqDataSource, XmlDataSource, SqlDataSource, QueryExtender, EntityDataSource

This objective does not include: AccessDataSource, SiteMapDataSource

- Query and manipulate data by using LINQ.

This objective may include but is not limited to: transforming data by using LINQ to create XML or JSON, LINQ to SQL, LINQ to Entities, LINQ to objects, managing DataContext lifetime

This objective does not include: basic LINQ to SQL

- Create and consume a data service.

This objective may include but is not limited to: WCF, Web service; server to server calls; JSON serialization, XML serialization

This objective does not include: client side, ADO.NET Data Services

- Create and configure a Dynamic Data project.

This objective may include but is not limited to: dynamic data controls, custom field templates; connecting to DataContext andObjectContext

Developing a Web Application by Using ASP.NET MVC 2 (13%)

- Create custom routes.

This objective may include but is not limited to: route constraints, route defaults, ignore routes, custom route parameters

- Create controllers and actions.

This objective may include but is not limited to: Visual Studio support for right-click context menus; action filters (including Authorize, AcceptVerbs, and custom) and model binders; ActionResult sub-classes

- Structure an ASP.NET MVC application.

This objective may include but is not limited to: single project areas (for example, route registration, Visual Studio tooling, and inter-area links); organizing controllers into areas; shared views; content files and folders

- Create and customize views.

This objective may include but is not limited to: built-in and custom HTML helpers (for example, HTML.RenderAction and HTML.RenderPartial), strongly typed views, static page checking, templated input helpers, ViewMasterPage, ViewUserControl

This objective does not include: Microsoft.Web.Mvc Futures assembly