

Sample Course Outline

| | |
|-------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| TITLE | RavenDB 4: Getting Started |
| LEVEL | Beginner |
| DURATION Estimate in hours | 2 hours |
| AUDIENCE PROFILE | This course targets developers familiar with C# and .NET who are looking at using RavenDB for their NoSQL document database platform and may already be familiar with SQL Server or relational databases used with ORM frameworks like Entity Framework. No prior knowledge of NoSQL is assumed. |
| ABSTRACT | RavenDB is a cross-platform NoSQL distributed document database designed for small to enterprise-level applications. In this course, RavenDB 4: Getting Started, you'll learn how to leverage RavenDB in the context of .NET application development. First, you'll learn what RavenDB brings to the table and what problems it will help solve. Next, you'll see how to create, update, and query data with the .NET client SDK. Finally, you'll discover how to manage Raven using the first-class "Studio" UI experience. When you're finished with this course, you'll have a foundational knowledge of Raven enabling you to use it in your next NoSQL-based application. |
| PREREQUISITES | The course assumes background knowledge of essential .NET Framework and C# with experience using Visual Studio 2013 or later. It also assumes some introductory knowledge of ASP.NET MVC concepts like Controllers and Actions but mainly for illustrative purposes. |
| DESCRIPTION OF SAMPLE PROJECT / SCENARIO | <p>I intend to teach the basic concepts of RavenDB using a prebuilt sample that uses ASP.NET Core to manipulate a database of TED talks. No JavaScript experience will be necessary or presumed, just introductory knowledge of ASP.NET MVC patterns.</p> <p>Code will be shown in C# in isolation (i.e. without any ASP.NET-related code) similar to a unit test. The result of changing the code to use Raven will be shown visually in the UI for the web app. This is to provide a</p> |

more visually interesting experience for the learner vs. writing text out to a console app.

The web app will have the following pages:

- **Homepage:** List of talks with speaker names, tags, descriptions. Sort the results, filter by criteria (speaker, year).
- **Edit:** Edit talk details using a simple form.
- **Search:** Search by keyword, sort the results by year. Same display format as homepage.
- **Watch:** Embeds the video to view in the page, show talk details, show “related” videos (based on tags)

The app will just use an in-memory sample dataset that matches the schema of the full dataset. As I teach a concept and part of the Raven client API, we’ll replace the in-memory methods with methods that use Raven. Users will be able to download the fully built app and it will have folders that incorporate the course code for each module.

All the business logic will be in a single service class (RavenTalkService) for simplicity and will be isolated by method so the viewer can focus on only the code that pertains to the problem at hand.

PLATFORM/TOOL VERSIONS

| Technology | Version(s) | Pre-release? (Y/N) |
|--------------|------------|--------------------|
| RavenDB | 4.0 | Y |
| ASP.NET Core | 2.0 | N |
| .NET Core | 2.0 | N |

RavenDB 4 Client API has very high backwards compatibility to 3.0+ clients. The Raven 4 Studio experience is largely the same but has a newer interface with more features.

Module Outline

1 Introducing RavenDB

22:00

Concepts:

Provide a brief overview of RavenDB, document databases, its features, what problems it's trying to solve, and how it compares to relational databases like SQL Server. Explain the dataset and web application we will be using to explore Raven's features. Explain how to clone the sample repository and install the .NET Core SDK. The developer should come away being able to explain how Raven is different and how it meets (or doesn't meet) their requirements of their application. They should also understand what topics and concepts we'll cover in the rest of the course, their local machine should also be set up to run the sample app to follow along.

Clips:

- Course Introduction (04:05)
- RavenDB Overview & Features (03:50)
- Raven vs. Relational Databases (04:24)
- Thinking in Documents (02:24)
- When to Use Raven(01:30)
- Sample Application Overview (04:34)
- Summary (00:43)

Demo Description:

Show the application UI that will be used to explore Raven's features. The demo will be the completed version of the app in order to show completeness and provide an example of what to expect. The app will have a list-based dashboard, a details page, a search page, and an edit view. Demonstrate how to download exercise files or clone the repository with Git and install the .NET Core SDK. Install Visual Studio Code and open the folder.

Assessment Questions:

- Which of these applications would be a good fit for Raven?

| | | |
|-----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| | <ul style="list-style-type: none"> - A stock brokerage application that needs to always be 100% accurate and deal with many different types of transactions - A media site that displays articles, comments, and needs to support a large amount of traffic across the globe - An analytics processing job that performs heavy aggregation of data for computational purposes - A Line of Business application used to manage inventory that is okay to be somewhat stale - A proxy application that caches information and retrieves it to handle tens of thousands of requests per second <ul style="list-style-type: none"> - What is the typical convention of the document key? (String prefixed with collection, e.g. Users/1) - Which aspect of Raven is ACID compliant? (Document operations) | |
| <p>2</p> | <p>Installing and Using Your RavenDB Environment</p> <p>Concepts:</p> <p>Guides the developer through installing a local development server of RavenDB so they can get started with the sample application. Will cover how to run Raven 4 on Windows and Linux in the most common way. Get a quick tour of the Studio interface to perform common tasks like viewing documents, editing documents, or creating new ones as well as viewing metadata. Run the sample application to make sure it works.</p> <p>Sections:</p> <ul style="list-style-type: none"> ● Overview (00:30) ● Installing the standalone server (03:13) ● Docker overview (01:40) ● Installing with Docker (3:00) ● Managing via command-line (01:30) ● Touring the Studio (08:00) ● Summary (00:37) <p>Demo Description:</p> <p>The demo will start by showing where to go to get the</p> | <p>00:16</p> |

| | | |
|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------|
| | <p>installers for RavenDB. For Windows, we will walk through downloading the zip file, extracting it, and running the server from the command line. For Mac/Linux, we'll install Docker and run the RavenDB docker image to host the database.</p> <p>We will open the Studio interface and walk through the basics of editing existing documents and creating new ones by using the "create sample data" Northwind dataset.</p> | |
| <p>3</p> | <p>Storing, Retrieving, and Saving Documents</p> <p>Concepts:</p> <p>The developer will learn common Raven operations that deal with the DocumentStore and DocumentSession. Learn how to retrieve and update documents in Raven using Load, Store, Delete and SaveChanges. Briefly cover Change Vectors and the semantics of transactions in Raven so that the developer is aware of possible design decisions later on in dev lifecycle. Raven is "safe by default" so we'll also cover common things the developer will run into using Raven day-to-day.</p> <p>Sections:</p> <ul style="list-style-type: none"> ● Overview (1:00) ● Configuring Visual Studio Code for C# and .NET (00:45) ● Connecting Using a Store and Sessions (04:00) ● Document Keys and ACID Compliance(2:00) ● Storing and Loading Documents (04:00) ● Loading Documents by Prefix ● Updating and Deleting Documents (04:00) ● Handling Concurrency and Conflicts (05:00) ● Conventions That Keep You Safe by Default (05:00) ● Summary <p>Demo Description:</p> <p>We will run the sample application and make sure it is loading the database correctly (if not, cover common issues–port, anonymous access). The demo app is designed to have functionality filled in so we will look at the aspects of the app that are missing the database code and fill them in with the appropriate commands. Creating, editing, and deleting a talk.</p> <p>Learning Assessment:</p> | <p>00:22</p> |

| | | |
|---|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------|
| | <ul style="list-style-type: none"> - How would you eagerly load an associated document? <ul style="list-style-type: none"> - <code>session.Load<Customer>(customerId).Include(c => c.Orders)</code> - <code>session.Include<Order>(x => x.Orders).Load<Customer>(customerId)</code> - <code>session.Include<Customer>(x => x.Orders).Load(customerId)</code> - In what cases does Raven guarantee ACID compliance? <ul style="list-style-type: none"> - Loading a document by ID - Loading a document where a property equals a value - Loading multiple documents that start with “Users/” - Querying an index by document ID - The recommended lifetime of the DocumentStore is: <ul style="list-style-type: none"> - Transient - Scoped - Singleton - LoadStartingWith returns potentially duplicate results: <ul style="list-style-type: none"> - True - How does Raven track changes to documents? <ul style="list-style-type: none"> - A simple timestamp field called a “change vector” - A version metadata field called a “change vector” - An ETag value - What method retrieves a change vector for a document? <ul style="list-style-type: none"> - <code>session.Advanced.GetChangeVectorFor(document)</code> - How many requests can you perform during a session before Raven throws an exception? <ul style="list-style-type: none"> - 30 - 40 - 32 - No limit | |
| 4 | <p>Querying Using Indexes</p> <p>Concepts:</p> <p>The developer will learn how to make basic LINQ queries and indexes. We will cover why indexes are important, Where queries, sorting, Map/Map-Reduce indexes, Side-By-Side Index Deployment, and querying via the Studio interface. This will cover the essentials any Raven developer needs to know to create a basic application.</p> <p>Sections:</p> | 00:29 |

| | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| <ul style="list-style-type: none">● Overview● How Indexing Works (05:00)● Querying and Paging with LINQ (08:00)● Query Stats and Customization (02:00)● Creating Map indexes (04:00)● Creating Map/Reduce indexes (04:00)● Searching and Field Indexing (03:00)● Managing Indexes in the Studio (00:08)● Summary● Course wrap-up <p>Demo Description:</p> <p>We will walk through implementing a basic search query against an index to make the search work. We will create indexes for Speakers and Tags so we can filter the list of talks. We will add queries to retrieve years and to allow filtering by year. We'll walk through the Studio interface to see how to inspect index definitions and how to issue ad-hoc queries against them. We will look at the performance timeline for indexes for debugging.</p> | |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

What's Not Covered

It's important to mention what is **not** covered in this course and would be better suited to a different course(s):

- Java/Python/Node.js SDKs
- HTTP API
- Loading document relationships in indexes
- Document and store listeners
- Document subscriptions
- Lazy and advanced queries (Lucene, RQL)
- Multi Map indexes
- Search & Facets
- Bulk database operations
- Clustering, replication and sharding
- Security/Hardening of server
- Voron overview
- Bundles (Unique Constraints, Versioning, Encryption, etc.)