Section 2: Creating a Serverless REST API

***Section Description (from the outline):*** In the second section you will learn how to create a REST API using Serverless. This API will use persistent storage to store data in an AWS database­ service called DynamoDB.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Video Number** | **Video Title** | **Problem / Solution (Not more than 50 words)** | **Step 1 (Not more than 10 words)** | **Step 2(Not more than 10 words)** | **Step 3(Not more than 10 words)** |
| 2.1 | Creating the core service for a Notes REST API | Create the base service and install all needed dependencies for a REST API. | Create a new service. | Install needed dependencies. | Install local development tools. |
| 2.2 | Adding a DynamoDB database connection | Explaining what DynamoDB is and configuring a database connection. | Adding the configuration in the handler.js. | Adding the configuration in the serverless.yml. | Configure DynamoDB locally. |
| 2.3 | Defining CRUD methods | Creating CRUD methods for the REST API. This video will show you how to stub out the main CRUD methods and create the events for triggering the lambda functions. | Add the function definitions and events in the serverless.yml | Add the stubs for all CRUD methods. | Testing the created endpoints. |
| 2.4 | Building logic for creating new notes | Writing the code for creating a new note and storing it in the database. | Adding database interaction to the create method. | Handling error and success responses. | Testing the new create method with a REST client. |
| 2.5 | Building logic for retrieving the created notes | Writing code for fetching notes from the database. Both fetching all notes and one note. | Creating a getOne method for fetching one note. | Creating a getAll method for fetching all notes. | Testing the new get methods with a REST client. |
| 2.6 | Creating an edit method to update a note | Writing code for editing an existing note in the database. | Creating an update method and adding database interaction. | Handling error and success responses. | Testing the new update method with a REST client. |
| 2.7 | Deleting the notes | Writing code for deleting an existing note from the database. Deploying the REST API to AWS. | Creating a delete method for deleting a note. | Testing the delete method with a REST client. | Deploying the REST API to AWS. |