

Ssample

using c# webservices add following data

user

account

Model

public class User

```
{  
    public int UserId { get; set; }  
    public string UserName { get; set; }  
    // Add other user properties here  
}
```

public class Account

```
{  
    public int AccountId { get; set; }  
    public string AccountName { get; set; }  
    public decimal Balance { get; set; }  
    // Add other account properties here  
}
```

Controller

[Route("api/users")]

public class UserController : ApiController

```
{  
    // Implement user-related methods here  
}
```

```
[Route("api/accounts")]

public class AccountController : ApiController
{
    // Implement account-related methods here
}

Action

public class UserController : ApiController
{
    private List<User> users = new List<User>();

    [HttpPost]
    public IHttpActionResult AddUser(User user)
    {
        // You can validate and add the user to your data store here
        users.Add(user);
        return Ok(user);
    }

    // Implement other user-related actions as needed
}

public class AccountController : ApiController
{
    private List<Account> accounts = new List<Account>();

    [HttpPost]
    public IHttpActionResult AddAccount(Account account)
    {
```

```
// You can validate and add the account to your data store here

accounts.Add(account);

return Ok(account);

}

// Implement other account-related actions as needed
}
```

With Db Connectivity

DB Structure

```
CREATE TABLE Users (

    UserId INT PRIMARY KEY IDENTITY,

    UserName NVARCHAR(50) NOT NULL,

    Email NVARCHAR(100) NOT NULL

);

CREATE TABLE Accounts (

    AccountId INT PRIMARY KEY IDENTITY,

    UserId INT,

    AccountName NVARCHAR(50) NOT NULL,

    Balance DECIMAL(18, 2) NOT NULL,

    FOREIGN KEY (UserId) REFERENCES Users(UserId)

);
```

APIs

```
[Route("api/users")]
```

```
[ApiController]
```

```
public class UsersController : ControllerBase
```

```
{
```

```
    private readonly YourDbContext _context;
```

```
    public UsersController(YourDbContext context)
```

```
    {
```

```
        _context = context;
```

```
    }
```

```
[HttpPost]
```

```
public IActionResult AddUser(User user)
```

```
{
```

```
    _context.Users.Add(user);
```

```
    _context.SaveChanges();
```

```
    return Ok();
```

```
}
```

```
}
```

```
[Route("api/accounts")]
```

```
[ApiController]
```

```
public class AccountsController : ControllerBase
```

```
{
```

```
    private readonly YourDbContext _context;
```

```
    public AccountsController(YourDbContext context)
```

```

{
    _context = context;
}

[HttpPost]
public IActionResult AddAccount(Account account)
{
    _context.Accounts.Add(account);
    _context.SaveChanges();
    return Ok();
}
}

```

JAVA

```

import javax.ws.rs.*;
import javax.ws.rs.core.MediaType;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;

@Path("/api")
public class UserService {

    private static final String DB_URL = "jdbc:mysql://localhost:3306/your_db";
    private static final String DB_USER = "your_username";
    private static final String DB_PASSWORD = "your_password";

```

@POST

@Path("/user")

@Consumes(MediaType.APPLICATION_JSON)

```
public Response addUser(User user) {  
    try {  
        Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);  
        String insertUserQuery = "INSERT INTO users (username, email) VALUES (?, ?)";  
        PreparedStatement preparedStatement = conn.prepareStatement(insertUserQuery);  
        preparedStatement.setString(1, user.getUsername());  
        preparedStatement.setString(2, user.getEmail());  
        preparedStatement.executeUpdate();  
        conn.close();  
        return Response.status(Response.Status.CREATED).build();  
    } catch (Exception e) {  
        return Response.status(Response.Status.INTERNAL_SERVER_ERROR).entity("Error: " + e.getMessage()).build();  
    }  
}
```

@POST

@Path("/account")

@Consumes(MediaType.APPLICATION_JSON)

```
public Response addAccount(Account account) {  
    try {  
        Connection conn = DriverManager.getConnection(DB_URL, DB_USER, DB_PASSWORD);  
        String insertAccountQuery = "INSERT INTO accounts (account_number, balance) VALUES (?, ?)";  
        PreparedStatement preparedStatement = conn.prepareStatement(insertAccountQuery);  
        preparedStatement.setString(1, account.getAccountNumber());
```

```
    preparedStatement.setDouble(2, account.getBalance());  
    preparedStatement.executeUpdate();  
    conn.close();  
    return Response.status(Response.Status.CREATED).build();  
} catch (Exception e) {  
    return Response.status(Response.Status.INTERNAL_SERVER_ERROR).entity("Error: " + e.getMessage()).build();  
}  
}  
}
```